Telektronikk

2/3.2001

Internet Traffic Engineering

Telektronikk

Volume 97 No. 2/3 – 2001
ISSN 0085-7130

Editor:
Ola Espvik
Tel: (+47) 913 14 507
email: ola.espvik@telenor.com

Status section editor:
Per Hjalmar Lehne
Tel: (+47) 916 94 909
email: per-hjalmar.lehne@telenor.com

Editorial assistant:
Gunhild Luke
Tel: (+47) 415 14 125
email: gunhild.luke@telenor.com

Editorial office:
Telenor Communication AS
Telenor R&D
PO Box 83
N-2027 Kjeller
Norway
Tel: (+47) 67 89 00 00
email: telektronikk@telenor.com

Editorial board:
Ole P. Håkonsen,
Senior Executive Vice President.
Oddvar Hesjedal,
Vice President, R&D.
Bjørn Løken,
Director.

Graphic design:
Design Consult AS, Oslo

Layout and illustrations:
Gunhild Luke, Britt Kjus (Telenor R&D)

Prepress and printing:
Optimal as, Oslo

Circulation:
4,000

telenor

Contents

Internet Traffic Engineering

Section I: Introduction/Overview

Section II: Traffic, Resources, Routing

Section III: Interdomain, SLA, Policy, Management

Section IV: Measurements

Section V: Systems and Services

Guest Editorial

TERJE JENSEN

Terje Jensen

To survive in today's competitive environment, a service provider must continually evolve its network and enable new revenue-generating services faster and more cost effectively than the competitors. Prior to the Internet, prior to recent mobile services, prior to e-business, a change in the telecommunication industry was seen as more predictable. Business leaders knew to take action – actions like reducing costs, launching new products, upgrading the networks, and so forth. Now providers are far less sure who their competitors are, the value of their core strengths and skills, and whether the business they have done well in for many years will continue to keep them profitable in the future.

Some may claim that a main cause for the uncertainty is that recent development of applications and service demands has been going on outside the sphere of the service providers. In particular, the Internet Engineering Task Force (IETF) has received major contributions from some main players and been guided by inputs from the academic world. For sure this has resulted in a plethora of applications and usage patterns, and phenomenal traffic growth. However, as more commercial concerns are entering the stage the providers would regain more control, for example by utilising the *Traffic Engineering* solutions. This also advocates further work on standardisation, ensuring interoperable configurations. Including procedures for managing multiple service types and requirements, Internet Protocol (IP) Traffic Engineering thus provides mechanisms for optimal operation and management of the IP-based network. Thereby, a provider would also improve its chances in the frenzied market.

Basically, one option could be simply to increase the capacity of the network, like adding more bandwidth to the links. A problem with this argument is that capacity should then be added wherever there is a problem, including the processing capacity, and also in the access network, on the servers, etc. Furthermore, and perhaps an even heavier argument is that the possibility for service differentiation would then still be rather limited. Being able to offer a portfolio of different services is recognised as a key enabler for ensuring a provider's profitability. Again, Traffic Engineering is promoting a set of mechanisms and procedures supporting a provider to achieve such goals. This becomes more important as the number of users and services grows.

The Internet Protocol (IP) has become a pivotal component in communication between various devices. It is rarely possible to make a single protocol suffice the diverse needs of all applications and users. To a certain extent, however, one may claim that the IP suite is addressing such an objective. However, looking at the original use of IP when it was designed, there are many other applications of the protocol these days; as more demanding services – like telephony, video distribution and mission-critical business applications – are gradually put onto IP-based networks, additional functions must be implemented in the networks and end-systems. Hence, one is stretching the capabilities of IP and additional mechanisms are necessary to allow IP to hold on to a central position. Several of these mechanisms are related to Traffic Engineering, that is, means activated to ensure the performance of the communication solutions. This will also allow for more predictable responses on service requests and swiftly support of more advance services and users.

Quite a few phenomena influence the evolution of IP-based networks, of which some interacting factors are:

- Increased load and expansion; more efficient ways of handling the traffic is sought. Scalability challenges are commonly faced for this reason.
- New technologies; efficient ways of interacting with IP-based networks are looked for. An example of this is the relation between functions related to IP and an underlying optical layer.
- New user groups; additional requirements could be placed on the IP-based network. Thus, efficient ways of differentiation between the groups are asked for, also accompanied by appropriate charging solutions.
- New applications; innovative ways of utilising IP-based networks are steadily observed, e.g. related to electronic business, mobile services, and so forth.
- Increased dependency on the network; coming from the service providers themselves as commercial aspects, but also from their customers, of which several are basing their business on an operational network.

Front cover: The conflict between market requirements and the service provider's resources

At low traffic pressure incoming traffic equals traffic carried. The artist Odd Andersen indicates this situation by his 45 degrees black lines. However, every traffic machine – be it switches, routers or the whole network – has a capacity limit at which the line breaks into a horizontal position. After that no more traffic can be carried whatever the pressure!

But the enormous increase – and non-established nature – of Internet traffic blasts through all aspects of capacity and quality constraints. The artist's skyrocketing white lines indicate the unprecedented demands to proper engineering of traffic machines in the new market environments.

The artist's generic message: Matching capacity of all components to uncertain Internet traffic demands.

Ola Espvik, Editor in Chief

- Increasing number of actors involved in service provision and delivery; connecting systems and networks managed by different actors, partly co-operating and partly competing, require adequate sets of means for a given actor to ensure its business and service levels towards its customers. This is further complicated by the dynamic commercial and technical environment that an actor faces.

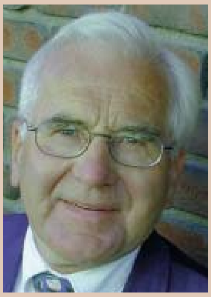
Several aspects have to be addressed as part of Traffic Engineering. A selection of the topics has been included in this issue of Telektronikk. These are divided into a number of sections as shown in the table of contents. Firstly, a set of papers of *introductory* nature presents an overview of the IP suite, history of Internet and principles of Traffic Engineering. Basic topics of designing and operating an IP-based network are then treated in a set of articles in the second section, called *traffic, routing, resources*. *Interdomain, SLA, policy and management* is the following section, addressing essential questions for commercially offering services – in a fast, accurate and automatic way. The papers deal with internal procedures and systems (e.g. management systems) as well as relations with other actors (e.g. agreements). *Measurements* are pivotal to follow and document the performance of the network. A set of papers is showing how to carry out measurements and factors to consider. The last section is called *systems and services*, presenting a few areas where solutions for IP and Traffic Engineering are used or required, like for mobile, for optics, and for voice.

As the use of *abbreviations* flourishes, a common list is collected at the end of this issue.

Computers and Communication

Early Development of Computing and Internet-technology – a Groundbreaking Part of Technical History

YNGVAR LUNDH



Yngvar Lundh (69) graduated from the Norwegian Institute of Technology in 1956. He served as leader of development teams and projects in computer and telecom technology, with the Norwegian Defence research Establishment until 1984, then with the Norwegian Telecom Administration / Telenor until 1996, then in his own consulting company. Since 1980 he also served as Professor of Informatics at the University of Oslo. His main projects concerned digital computing and circuit technology, enabling the start of "hi-tech" companies, notably Norsk Data AS, and systems for defence and for telecom enhancing and employing emerging technologies. Lundh is a member of he Norwegian Academy of Technological Sciences.

yngvar@joker.no

Building the Arpanet and using it as a laboratory for development was a major contribution both to computing and to telecom. During the 1970s a very innovative and thorough research and development took place under the leadership of the United States Department of Defense, Advanced Research Projects Agency. From the outset the effort can best be characterized as basic technical research. Ten research groups worked together as a team at developing the Internet-technology itself. At the same time the development was influenced by many other research groups, mainly in the United States, who actively pursued specialized networking applications, thereby creating needs and uncovering possibilities. Concurrently dramatic developments took place both in circuit and device techniques for computers and in the politics of telecom operation. A glimpse of the Norwegian perspective is included through the eyes of the small participating group in Norway. The text endeavours to be readable without prior technical special knowledge. Its objective is to interrelate main technical events of computing in a historic perspective.

Strengthening Telecom and Computer Techniques

Internet is not only a modern hobby, nor an exotic new trading arena, nor new mail, nor new distribution of information. Internet is all of that and much more. A result of a successful development since the late 1960s of basic new technical principles, it pertains to *computer collaboration* and to various forms of *information transfer*. We shall refer to these techniques as Internet-technology. During the same period – the last 25 years of the twentieth century – important development took place in two other areas. Electronic *circuit and device techniques* made it possible to fabricate computers cheap and small, thereby making them more interesting economically. They can now be used as components in new roles (for that matter the same circuit technology also made even more powerful computers possible). *Telecommunications* went through a profound *reorganization*. This admitted new actors and driving forces into the technical and commercial evolution of telecom networks.

Those are features of a development presently causing drastic changes in our relationship with information and its use. Most likely we are just at the beginning of a new era. Nobody knows where it will take us. This technology is already beginning to change important functions of our society. Some of the perspectives that can now be perceived imply exciting new possibilities.

This article describes that technical development without assuming special technical knowledge and avoids detail that is better described elsewhere. Outlines of the historic development of the internetworking technology and parts of computer- and telecom techniques are described. Special focus is directed at the 1970s when most of the basic technical development took place.

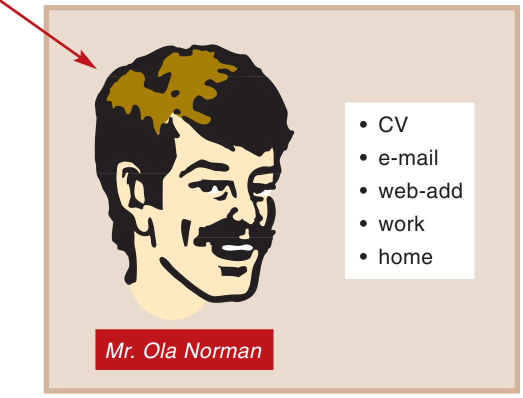
World Wide Web – WWW

People today tend to think of Internet as synonymous with World Wide Web. That technique opens a splendid new view of a world of information. Simple point and click movements catch posted information anywhere in the world independently of distance. Information is posted in the form of "home pages". The information is coded in standardized format and is stored in computers connected to the network. For the coding and formatting various aids are readily available and easy to use. Hence the "Web technology" can be used by anyone who wants a message to be presented. It immediately becomes available to the whole world. That is a world that can be described as "countless millions" and which grows such that it will shortly comprise at least everyone who has or could have a telephone. At the same time the technology is developed further towards user-facilities with far more comprehensive abilities than telephones at conveying information.

However, World Wide Web is far from the entire Internet. It is only one – admittedly very conspicuous – example of exciting new possibilities that can be implemented on top of the Internet-technology itself. The Web technique first emerged as a practical internal information distribution system in a large international research establishment in Switzerland – CERN – in 1990–91. From there it spread incredibly fast throughout the Internet, i.e. to "the whole world".

But the basic ideas were already 25 years old then. The idea of being able to open windows via CRT-screens on collections of information of various types near or far and to navigate in a world of information by point and click movements of hands and fingers were demonstrated already in 1968.

-----In any picture or text hyperlinks may be built in. As an example, clicking the name of a particular person may immediately bring in his picture located in some computer – anywhere in the world



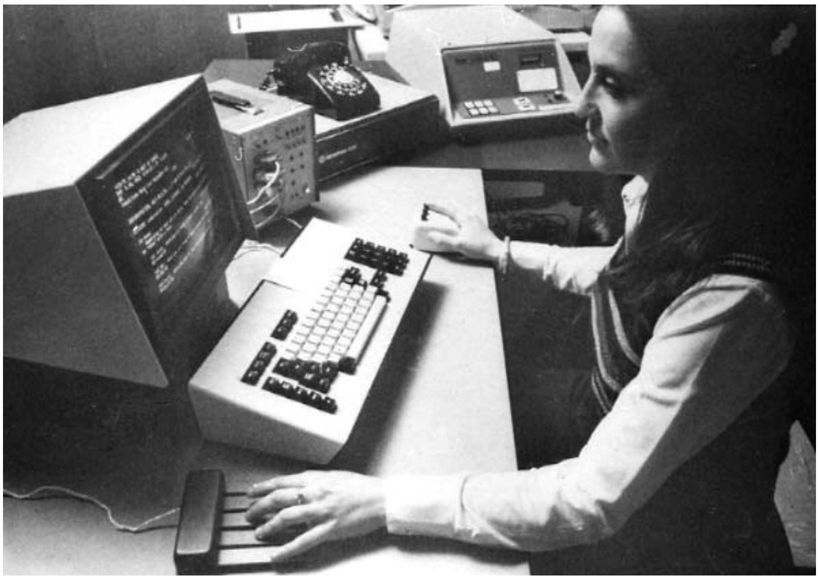
Hyperlinks make information stored elsewhere available by pointing at a reference to it and clicking

The leading person and creating and driving force in these developments was Douglas Engelbart of Stanford Research International – SRI – in Menlo Park, California. Engelbart developed and demonstrated a comprehensive set of concepts and techniques of groundbreaking importance. The most significant was perhaps the "mouse", the little thing you hold in your hand. Everybody who has seen a computer today has seen it. Equally important is the "hyperlink". That is a reference pointer-code that can be embedded in any information-image – text or picture – displayed on a computer screen. A mouse-click on the pointer opens the referred image. That happens independently of where in the net that information happens to be located.

A quarter of a century would elapse before these ideas could be seriously employed and put to extensive use. Only then the technical and economic prerequisites were met. Microprocessors had made low priced personal computers common as "workstations" and the Internet was available and ubiquitous (networking workstations were another of Engelbart's ideas long before the microprocessor). And, of special importance, the ban on commercial traffic in the Internet was lifted in 1991. Some of the prerequisites were met relatively early. Computer screens, first demonstrated in the late 1950s, began to be common from the early 1970s, personal computers a little later. Computer co-operation in general, vendor independent networks, beyond the groups participating in developing the Internet techniques themselves, became usual from around 1980. Some of the ideas that Engelbart demonstrated in 1968 are still (2001) awaiting comprehensive use, but are expected to become similarly important. Examples are telephony and moving pictures.

World Wide Web is primarily an example of technical possibilities that Internet technology opens up as a carrier of entirely new and exciting forms of information handling.

The concept of workstation was demonstrated at SRI by Douglas Engelbart as early as 1968. It had cathode ray screen, a "mouse" for the right hand and a five-finger key-set for the left hand. The user could point, click and type anywhere at the screen while looking steadily



Electronic Mail

Message transfer was a dominating application of the Internet already from the start of the development in 1969. That form of communication was especially practical and a necessary tool in the decentralized collaboration of the ten groups of researchers who undertook the basic research and development in the 1970s.

The original vision of resource sharing networking, an important source of inspiration for the development, comprised a number of other, more or less exotic, applications. E-mail was an overwhelming generator of traffic for many years. It perhaps still is, at least was so until the Web started another "landslide" of new users. To many people either e-mail or Web is still synonymous with the Internet.

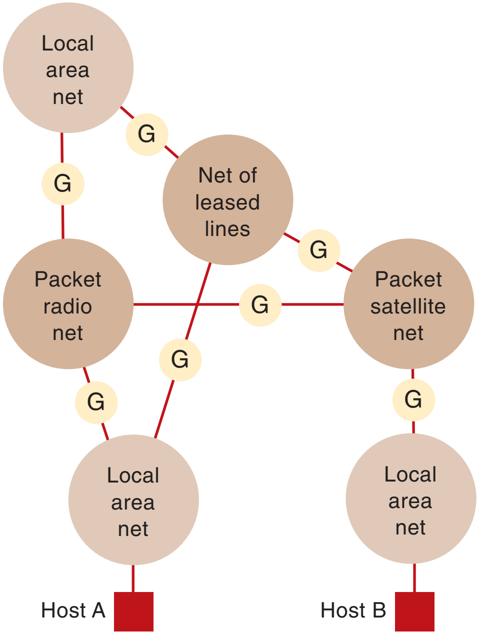
E-mail is an important form of communication already. It has unique properties in comparison to ordinary mail and telephone. Therefore it defends a place of its own as a communications medium. It overcomes both time and distance, it is fast, but still the addressee may answer precisely, for record, when convenient after having had plenty of time to think, unlike the telephone. E-mail is suitable for automation in various forms. And it is cheap.

However, e-mail is still – in 2001 – far from having reached its full potential as a general communication medium. "Old-fashioned" mail and telephone are far ahead in general availability. The most important shortcoming is that e-mail only reaches those who often use computers, connected to the Internet – and *when* they use the computer. To make certain that a message gets there, at least to most people, it will still be best to call or send a letter.

E-mail will be much more usable the day when the receiving apparatus immediately emits a beep or blinking light signalling the arrival of a message. The messaging apparatus should be equally cheap to own as a telephone that rarely rings. We are still some distance from being able to send important messages to many recipients by e-mail *only*. But the trend is pointing in that direction. And a transmitter and receiver of e-mail will not be a PC only. It will be built into telephones or other future "popular boxes", that we have not yet seen, but which will creatively be made usual in the future world. Many such boxes will be gadgets with remotely controlled functions, etc.

Wireless or Wired Somehow

The actual reason for the name Internet is that the network may employ various carrier media of many different kinds interconnected for the transportation of information.



The Internet is a network of nets interconnected by gateway computers. Nets may be of different types. From one host computer to another information travels in packets along routes which may change with network "shape" and traffic load. The Internet Protocol (IP) helps navigate through the network. The Transport Control Protocol (TCP) helps ensure that a message transported piecemeal as a number of packets gets properly re-assembled

Further, Internet-technology implies mechanisms for optimized mixing of different transport requirements. Urgent messages get there fast while less urgent traffic may be transported more cheaply using otherwise idle periods. Traffic types are more or less error prone. Particularly important traffic needs precedence – e.g. for resolving problems in the network itself, and so on.

Characteristics such as error density, urgency, and precedence are measurable quantities to be specified, and to be met accordingly. Internet technology lets different requirements be met automatically by the available network capabilities.

In particular this enables many carrier media of rather different capabilities to co-operate in the transport such that each medium is exploited to its best ability. Prevailing media now are leased lines capable of specific bit transfer rates – number of pulses per second. Various standard bandwidths (pulses per second) are available. Several other carrier media are important and will be used increasingly. Local area networks prevail within buildings and geographically limited corporate sites. Radio computer networking of various kinds is evolving further.

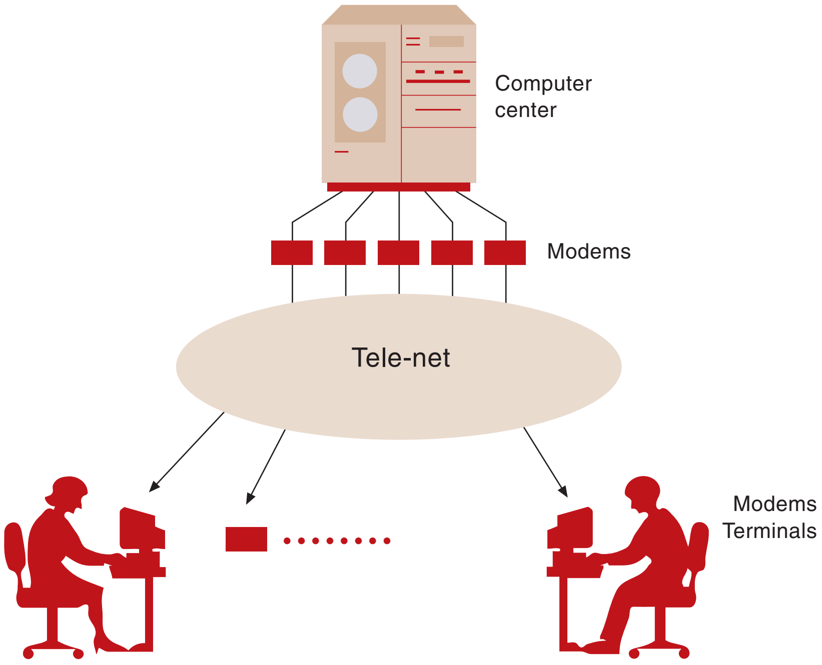
Satellites have many exciting possibilities. The same is true of cable networks originally built for broadcasting television. These are probably best suited for rural and urban areas respectively, and have great potential for being exploited further.

The background for the name Internet is *net of interconnected nets*. The individual transport networks are operated separately as mask-shaped nets of leased lines, packet radio nets, packet satellite nets, and so on. Each of these media transports packets in ways best suited for each type of net. The individual nets are connected into one network (of nets) called *Internet*. Gateway computers make the interconnection. They convert the information on its way to the next net into a format suitable for appropriate handling there.

The basic development of Internet-technology took place in the period 1969 – 1980. It comprised understanding the problems and the possibilities, creating and testing the technical methods and defining the results as standards that were open and available for use by anyone.

The main result of this development is that different computers can now co-operate and exchange all types of information. Further, the

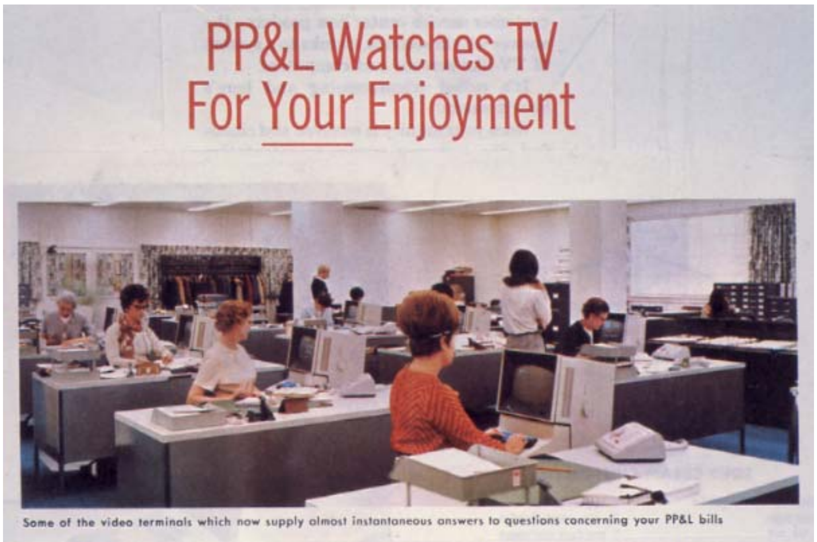
Teleprinter, ("Teletype", "Telex machine") built for the international Telex network was extensively used as computer terminals



Timesharing allows many users to work directly and simultaneously with the computer from remote locations

information can be transported through different types of transport media. Each transport is automatically handled as required by interconnected carrier media that may have different characteristics.

The transport is handled according to technical rules called protocols. They depend on more detail and are significantly more complicated than the protocol that govern ordinary telephone traffic. In reality the protocols are implemented as programs in computers. Microprocessors of various processing powers are now available and can be used as components in devices built to use the Internet for various purposes. It appears that we are presently at the beginning of an era of future types of information networks.



PP&L Watches TV For Your Enjoyment

Some of the video terminals which now supply almost instantaneous answers to questions concerning your PP&L bills

Figure 5 CRT-screens and timesharing were major steps in adapting the power of computers to the abilities of people. This picture was copied from a brochure in 1971. The large company impressed customers calling in. As soon as the customer gave his name the operator could (actually type it immediately and thus display his data, hence –) "remember" details about him

Remote Computers and Time Sharing

Early computers and their users communicated using teleprinters – "Teletype machines" – that conveyed text directly both into and out of the computers. For many years information was transferred in and out of computers via punched cards or punched paper tape. The need for efficient use of computer time dictated these media. They were much faster and hence occupied less of the valuable computer time waiting for slow fingers and printer mechanisms. Special line printers to be directly driven by the computer were developed. For many years powerful line-printer machines were essential parts of computer centres, and they produced vast amounts of paper printouts. Although they are capable of fast printing of text on paper lineprinters have been surpassed in performance by newer devices. Both ink jet printers and xerographic printing mechanisms using lasers for pattern generation are almost household items today, and they outperform the earlier, powerful mechanical line printers both at efficiency, quality and flexibility.

The first operating systems were developed early in the 1960s. That is programs that manage the computer itself and its attached resources. Since then a computer without an operating system is unthinkable. The operating system manages the computer and its various tasks. As an example the operating system permits the fast central processing unit to carry on at full speed while slower attached units such as printers work at *their* speed. Important operating systems today are Windows, NT, UNIX and Linux, each in several variations. While Internet-technology is independent of individual machines and operating systems, the one operating system that was most prevalent during the first decade of networking development was "Tenex" by Digital Equipment Corporation – DEC. It was a popular operating system especially for that company's tenth major computer model ("Programmed Data Processor") – PDP 10. Towards the end of the 1970s Unix became more and more popular and has become an important industry standard serving many types of computers. In the 1990s Microsoft Corporation's various "Windows" systems have overtaken it in volume outnumbering all others.

A remarkable phenomenon during the period of internetworking development is the absence of IBM from that development. From the late 1950s that large, internationally distributed company had a uniquely large market share for computers and everything in computing. Especially IBM took leadership in many areas of technical standard setting. Cards, magnetic tapes, codes, etc. were "IBM-compatible". IBM also set their own comprehensive and successful standards in computer networking.

The operating system enables a computer to do several tasks at the same time. A particularly important development was *timesharing*. The Computer Time-Sharing System – CTSS – was demonstrated in its earliest form at MIT in the early to mid 1960s. Timesharing permits several users to access the computer simultaneously. Several user terminals, typically Teletype machines, may be connected to the computer. Each user experiences the communication with the computer – via the timeshared operating system – as if she had the computer to herself alone. The computer actually shares its time between several users and several tasks. The users perceive increased load from more users as slower response from the computer. Powerful computers may serve many, perhaps several tens of users without noticeably slow response.

Special printer terminals became available

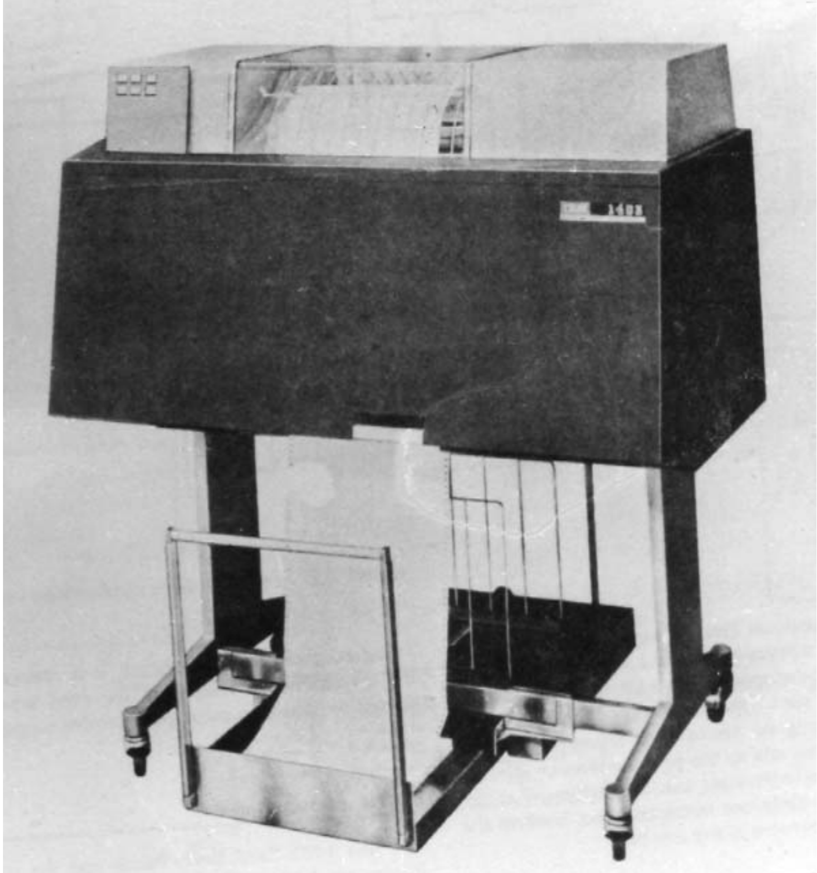
Standard Teletype machines are specialized typewriting machines made to be connected for transfer of written – teletype – messages through the international *Telex* network. In the 1960s specialized typewriting computer terminals began to replace the use of Teletype machines for computer purposes. They were faster and more flexible to use, had more comprehensive character sets and could produce nicer print. Only from the beginning of the 1970s terminals with Cathode Ray Terminal screens became more and more usual. From around 1980 CRTs were dominating computer use. Terminals are connected to computers via [telephone] lines. Gradually it became usual to employ the telephone network for communication with computers. The signals between the terminal and the computer were converted into signals that could be transmitted similarly to speech signals. They were "modulated" and "demodulated" by *modems*. In this way users of computers could have terminals placed at the users' premises, connected to computers elsewhere via fixed – leased – or "dial-up" telephone lines.

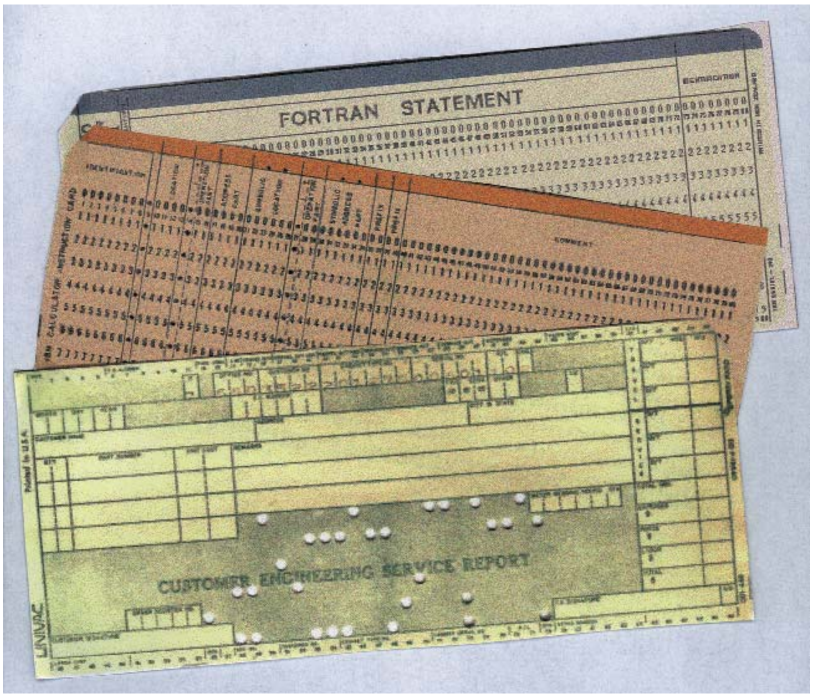
Vendors of computing services established large computer centres sometimes with extensive nets of leased lines and connection points for modem connections. The one single application making most use of early computer networking was seat reservation for airline passenger traffic. Already in the 1960s the large airlines had ubiquitous, often global networks for that purpose.

Computer Centres and Personal Computers

For economic management of expensive computers stable full employment of the machine is important. Before the invention of timesharing computer centres were run in "closed shop mode". The actual users, i.e. those who developed programs and those who delivered data for processing, were not allowed to communicate directly with the computer. The user submitted jobs consisting of programs and/or data, usually in the form of a deck of punched cards or a roll of punched paper tape. A popular profession was that of punching machine operator, transferring written programs or data from paper into punched cards, using the specialized "off-line"

The line printer was the output medium from computer centers. It produced large amounts of printed pages folded together

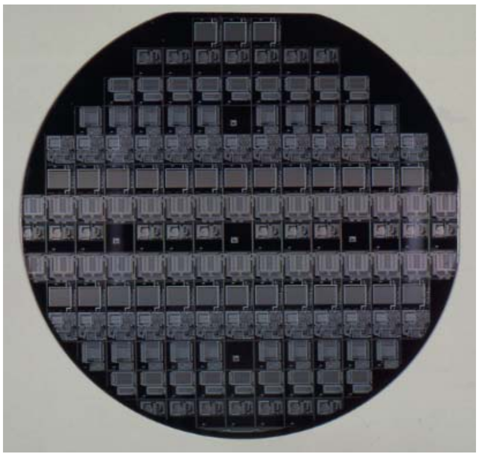




The punched card was an important input medium well into the 1970s. It was only replaced when timesharing and sufficient disk storage permitted users to enter their programs directly and leave them in the computer. One card represented up to 80 characters. (Pictures copied from "Britannica.com")

punching machines. Operating staff accepted jobs through a window and delivered results in the form of line printer print-outs (and the returned card decks). During program development that typically meant lists of error messages. The primary task of the operating staff was to keep the computer in stable and reliable operation for high productivity, referring to the use of computer time as an overwhelming cost factor. Those card decks could be large, sometimes filling long steel drawers almost too heavy to carry.

Electronic integrated circuits and devices for storage and presentation (semiconductor memory, storage disks and CRT screens) have gone through a very comprehensive development indeed continuously since 1960. Then the first digital circuits could be built in large scale using transistors rather than vacuum tubes. That development improved performance characteristics by many orders of magnitude (and continues rapidly to improve further).



By 1970 the integrated circuit technology had developed to a point where microprocessors could be "stamped out" cheaply on silicon wafers in large numbers. This picture shows an experimental wafer with an array of several different circuit chips. The matrix is cut into individual chips

From the mid 1970s that technical development permitted complete programmable computers to be made cheaply enough to make it meaningful economically for one person to have the computer alone. The movement of the human person's fingers and her ability to think and formulate commands and questions are very slow if measured in a computer's time scale. Looking at the exploitation of a personal computer – PC – one will typically find the computer running idle most of the time while the user thinks or does something else. The PC-phenomenon – economically seen – is that it makes economic sense to have the machine idling, but immediately available to the user – the person. Unlike the situation today that was far from true for many years.

Resource Sharing Networks

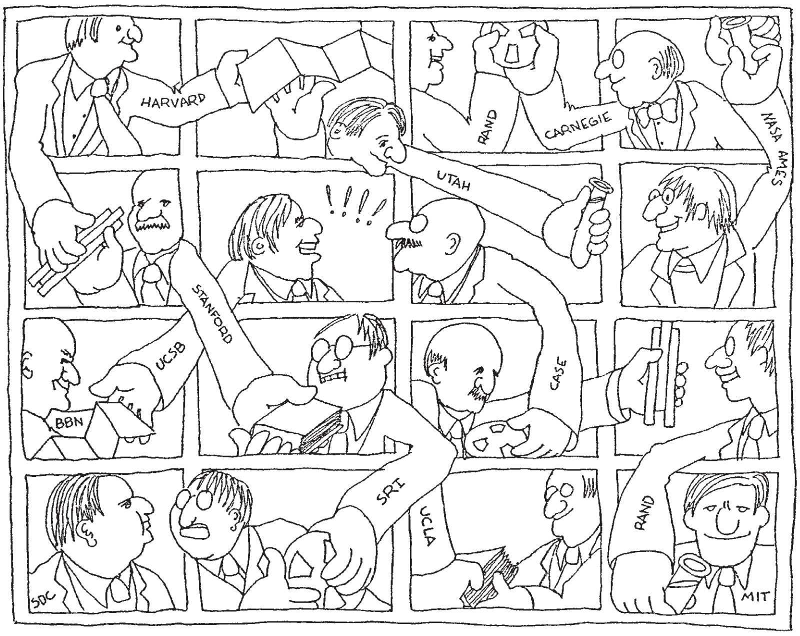
Computers began to have practical significance in business as tools and production machinery from the last half of the 1950s. Computer technology and its use have continued to improve and increase since then. This technology has an unbelievably large and increasing importance. The development has continued and continues on. All the time it becomes more specialized and refined.

Academic interest in this dramatic development started in a few places. From the late 1950s it was the basis of a new industry, rapidly growing. IBM was the foremost, gigantic, standard setting "powerhouse" among several great companies. A large and growing number of engineers and technical scientists engaged themselves in the study and development of the great new potential that they saw and that began to open up around 1960.

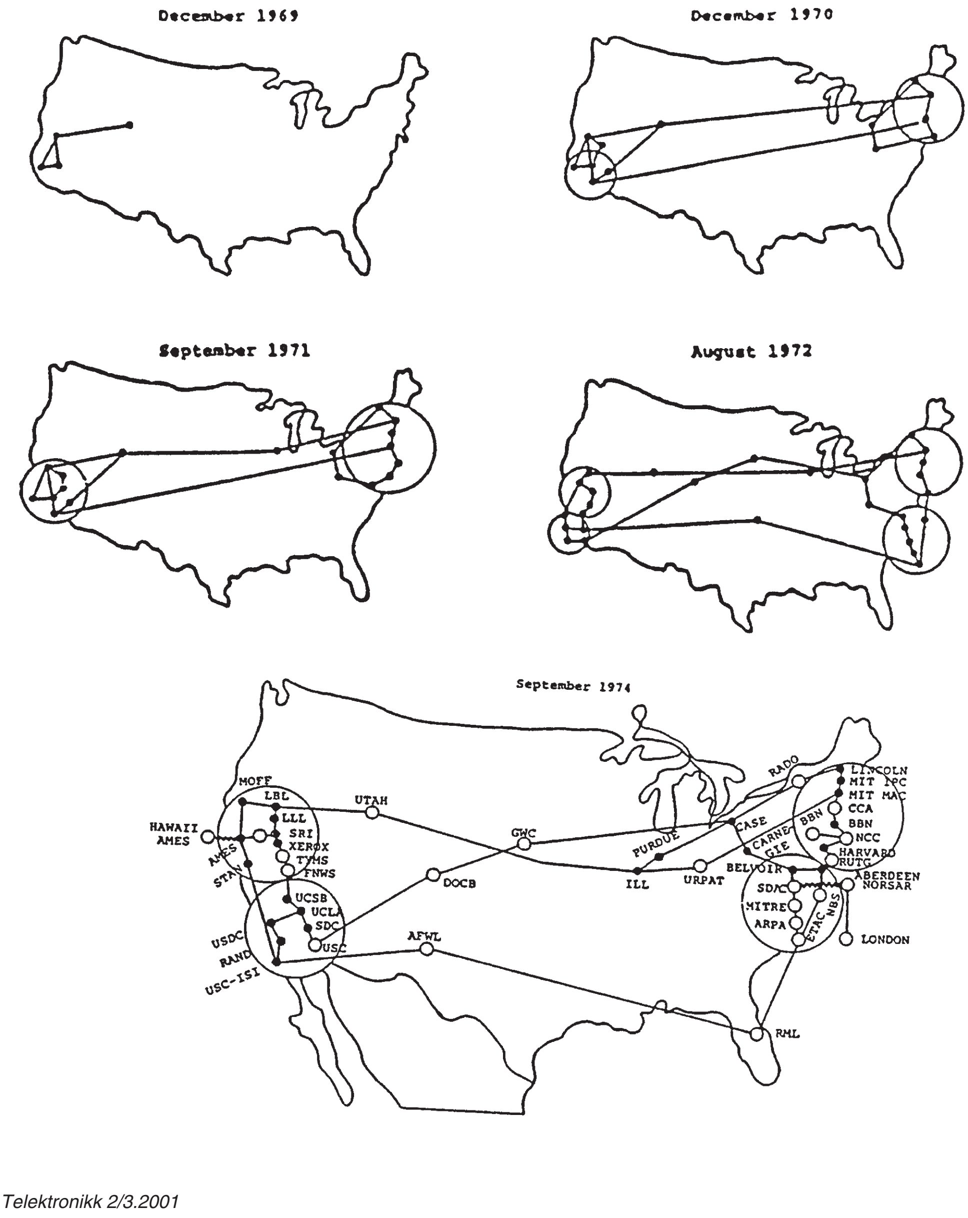
One broadly based research program started in the late 1960s. The Advanced Research Projects Agency – ARPA – of the United States Department of Defense sponsored the building of a resource-sharing network called *Arpanet*. It connected four computers at universities and research establishments in the western United States. The research program was basic technical research and the visions about Arpanet were *resource sharing*. Important and valuable resources that could be shared and hence be better exploited were several: Powerful computers (although quite weak by today's measures) were too expensive to be procured by all those who wished for them and who could have put them to good use. It became desirable to make such computer resources available to more groups of people. Thereby new problem areas might be attacked and manifold creativity might more easily meet in fruitful collaboration – resource sharing.

Arpanet grew fast and more academic groups in the USA were connected to the resource sharing network. The research and development concerned many technical and scientific areas and a wide spectrum of applications were pursued in addition to computer and networking techniques themselves. The numbers of people and research groups connected with the Arpanet far outnumbered the relatively few that worked on the basic Internet-technology itself.

Examples of applications being investigated and connected to the Arpanet as early as 1972 were weather forecast, funds transfer, understanding of natural language (e.g. natural question-asking), telephone conferencing, mathematical analysis, and others. The purposes of the use of the network were many. One example was an interactive program for mathematical analysis – “Macsyma” – at MIT. That program was available for use through the Arpanet for several years for anybody interested. Thus a great num-



Resource sharing networks. This drawing covered the program of one of the first conferences publishing the Arpanet effort



In 1969 Arpanet was first built as a packet switched computer network for resource sharing. Many academic groups in wide fields of research joined during the first years. By 1974 the Arpanet extended west to Hawaii and east to Norway and England ber of qualified mathematicians in universities were critical users while the development proceeded further. Errors and suggestions were easily reported to the developers. In this way the program was subject to brutal testing and consequent help in discovering weaknesses in the interactivity of the program, a new and very effective way of improving program robustness. And a free source of good ideas was available from a world of interested and competent users. Many other research projects in many areas proliferated around the network from the start.

Basic Open Technological Research and Development

Basic ideas and techniques for networking were implemented already in the initial "little" Arpanet in 1969. The networking technology was further refined and developed in an intimate co-operation of ten research groups during the 1970s. That co-operation resulted in the technology underlying today's Internet. The results were documented as standards. They were made available to the US defense to be further formalized and were openly available to anybody world-wide, especially interested academic groups.

That led to an increasing interest, also internationally, through the 1980s. From 1991 a very rapid growth began. The public did generally not know about the technology until 1995. The work was not secret, however. The work and the results were available to interested researchers from the start in 1969. The basic idea was resource sharing including human resources, ideas and suggestions. And much was done to create interest. A comprehensive public conference with demonstrations of several research efforts was held in Washington DC in 1972. Comprehensive presentations were made at the Computer Communication Networking Conference at Brighton, England in 1973 and at the International Computer Communication Conference – ICCC – at Stockholm, Sweden in 1974.

It was always essential for the development that any kind of computer of any make could co-operate in the network. Internet technology has become an important reinforcement of the foundations of both computer technology and telecom technology. In turn that has stimulated innovations in business and in society in many ways. Most likely we have just seen the beginning of that development.

Internet Details

What is Actually Internet

In technical terms Internet is a network connecting co-operating computers. The network transports information. The connected computers are called host computers. They exchange information with each other. The information transport and exchange takes place according to standardized technical procedures called protocols. Various quantitative requirements of speed and other factors can be specified for each transportation task. The transport network also comprises computers built into it, that carry out the logical functions of accepting, delivering and routing information being transported.

The name "Internet" reflects the fact that the transport network actually is a network of individual interconnected nets.

Computers can have various forms and sizes. A computer may also be small and cheap and may be built into an apparatus as a component, e.g. in a telephone. Telecommunications are expected to make increasing use of the Internet technology, which will thus have an increasing role in future telecommunications.

Experiments in Computer Networking

Historically some computer networking experiments began in 1969 as an experimental network called Arpanet. It consisted initially of a number of nodes connected by leased lines that could transmit digital data streams. Each node was a computer with a program that made it function as a so-called "Interface Message Processor" – IMP. The lines with modems could transmit up to 56,000 bits per second, somewhat more on some "legs" later on. One or more host computers could be connected to each IMP. Hosts might be of various types and sizes and be used for various applications. Unlike earlier computer networks the Arpanet distinguished between the IMP-computers that were part of the transport network, and the host computers that were the co-operating computers proper.

From the initial Arpanet the technology was developed into a basically new computer co-operating technology – *Internetworking-technology*. Its main constituents were defined around 1980.

Some further technical refinement and significant further geographical expansion of the network took place through the 1980s. This was all carried out on a non-commercial, experimental basis. The network spread into many countries around the Earth, primarily to universities and research groups.

Commercial traffic was prohibited until 1991. That exclusion was then lifted. From then on the growth of the network accelerated. In the early 1990s the growth corresponded to doubling the number of connected host computers every seven months approximately.

Network of Nets and Packet Switching

Information gets transported through the net in packets. The net is said to be packet switched. A packet consists of a certain number of bits – binary digits – e.g. 2048 bits plus a "packaging" of a few bits. Packet size (number of bits) may vary for different applications and different nets. This packaging of header bits and trailer bits specify how the packet is to be treated in the transport net, and identifies the packet to the receiver. Each digit assumes value 0 or 1. In transmission each bit is typically represented as "pulse or no pulse" at defined instants in time.

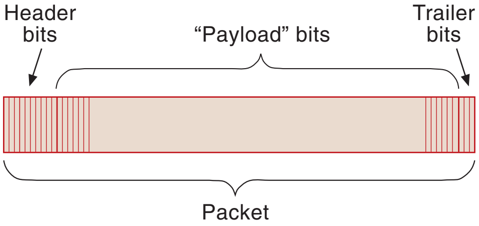
The initial Arpanet had permanent leased lines connecting the nodes. The further development comprised, among other things, possibilities for other types of transmission carriers. Special interest concerned packet radio nets where each node had a radio transmitter and -receiver and all nodes used the same radio channel – carrier frequency. The purpose was to exploit such information carrying media and their special properties. They have potential for connecting ships, aircraft, cars, persons, as well as more or less temporary platforms such as oil rigs and other stations in wayless territory or in developing areas having inadequate permanent installations. Further developments comprised ways of using satellite channels and special cable channels in similar ways.

Intensive basic research and development was carried out through the 1970s. It turned out early that the logical rules – protocols needed to exploit each carrying medium – were rather different from each other. An overriding goal was that each node should function independently without a common control centre.

It was discovered that advantage could be gained by keeping each carrier medium separate as an individual net, e.g. packet radio net, packet satellite net, local broadband cable net, etc. Gateway computers then interconnected each of these nets. Each gateway appeared as a host computer to each of the two nets that it connected, behaving according to the protocols of the individual nets. The gateway re-packaged each packet accordingly. Hence transfer could be done according to the individual network protocols and each net was exploited well.

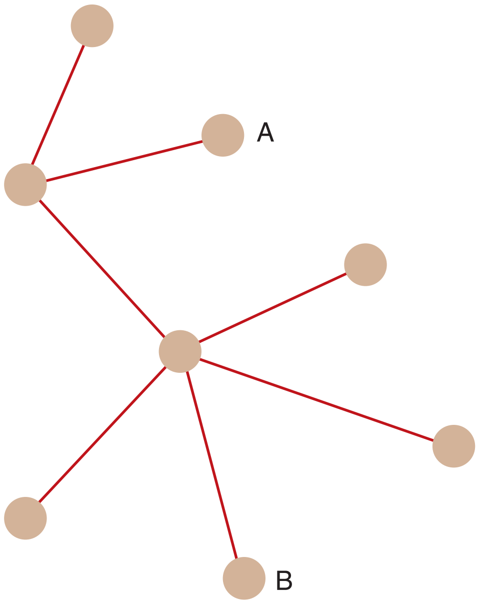
Stars and Masks

Many computer networks were built and operated, for practical commercial purposes, long before the Arpanet and the Internet. Notable are large networks for airlines, as mentioned. Private and public organizations had been using networks of computers since around 1960. Perhaps most notable were networks and associated standards developed and built by IBM.

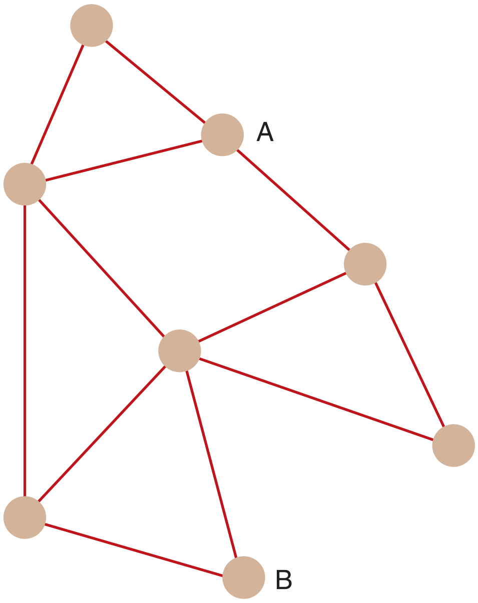


A packet switched network transfers streams of packets. Each packet has a limited amount of "payload" information plus a "packaging" of some header and trailer bits for addressing and transport specifications. Transfer of a packet over a line only occupies the line for the short time of that limited number of pulses

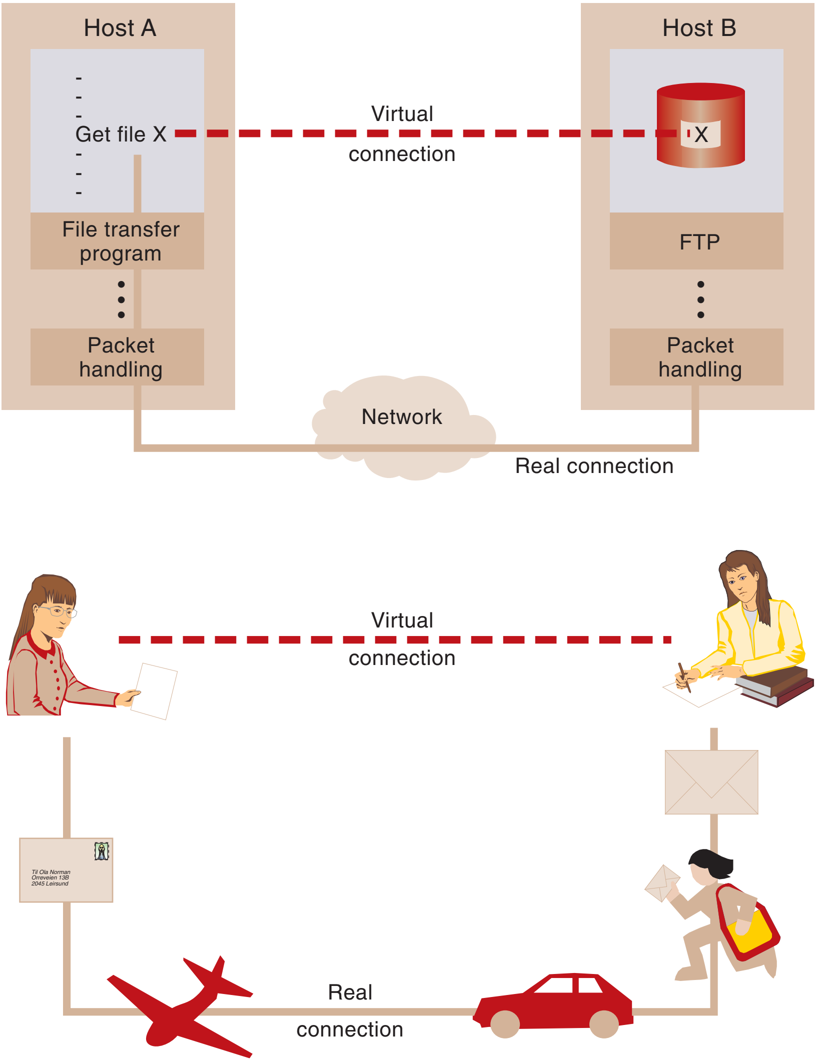
The first computer networks were mostly if not exclusively star shaped. The transfer channel between any two points A and B in a star shaped network is unique. The Arpanet/Internet development consistently made use of mask shaped nets. Transport between arbitrary points A and B in a network may then travel alternative routes. This strategy, well known in traditional telecom networks, has many advantages. In the more



Star shaped network



Mask shaped network



Layered protocols let application programs in co-operating computers talk "virtually direct". Actors (programs) at each level need not worry about details of what goes on at the lower levels of the information exchange. An analog situation is in correspondence by mail, where users put (and get) the enveloped letter in a letterbox and leave the many detailed procedures ("lower levels of protocol") to the various departments of the Mail

diverse and dynamic traffic situation of computer networks the enhanced reliability and traffic resilience of mask shaped nets are essential. A significant part of the development effort was concentrated on a desire to handle these needs of computer networking automatically and well. One issue in that development pertained especially to mask shaped – as opposed to star shaped nets. Both traffic control and reassembly of large messages are more complicated in the former. But with the resulting TCP/IP (Transport Control Protocol / Internet Protocol) protocol suite mask shaped nets can exhibit their advantages automatically in dynamic situations.

The result was the protocol suite TCP/IP. The successful, persistent and ubiquitous use of TCP/IP is now established in millions of nets of even more millions of computers. That deserves being mentioned explicitly. Those protocols resulted from an extremely thorough analysis and design. "No stone was left unturned" during the development which took several years. Theoretical analyses were complemented by experiments. Combinations of traffic types and requirements, network topologies and application types were imagined, implemented, tried, failed, changed and tried again. The "final" TCP and IP were not easily postulated and approved.

Traffic Diversity

Nobody can ever reproduce in a laboratory the chaotic traffic pattern of a lively telecom or computing network and even less the diverse demands of information exchange. The growing active dynamic traffic situation in the Arpanet prevailed during onwards development of its own underlying technology. That may be one reason for the robustness, elegance and survivability of the result. Arpanet was the laboratory. At the same time it was an active telecom network, a resource sharing network and a forum of creative and critical people. During a period of intensively active development methods were conceived and perfected until functioning well in an environment which was closer to reality than anyone might have dreamt up in a "sterile" laboratory environment. At the same time a profound theoretical understanding was developed. It kept its scrutiny on experimental results and was both guiding and following up the work in an admirable teamwork. The group at UCLA under Leonard Kleinrock's leadership was foremost in that area.

One of the many experiments illustrates some of this thoroughness, the *Internet conference speech* experiment. Digital speech coding is of course a long story in itself. Whereas international telephony today uses 64,000 bits per second to represent ordinary speech, many other methods can represent fully understandable speech by much more compact codes. One example is Linear Predictive Code – LPC. It permits understandable speech to be represented by 2,400 bits per second (this is not the limit, but it was used for that particular experiment). Such compact coding is less tolerant both of background acoustical noise around the speaker and of errors in the transmission channel. Loss of a packet is more harmful in the more compact codes. An LPC coder/decoder (codec) developed by MIT's Lincoln Laboratory was used for the experiment. The first version of the codec, a rack mounted unit so large it could barely be lifted by a man, was successively replaced by smaller and lighter units doing the same. It helped clearing the way towards possibilities of today's integrated circuit chip solutions.

Further about the conference speech experiment. A rather many-sided development had created solutions and understanding of protocol options, network configurations, packet satellite channel access algorithms and their inherent stability, various performance characteristics, and many other factors, before the final experiment. It demonstrated internetworking speech conferencing. Three persons located at Boston (Massachusetts), London (England) and Kjeller (Norway) held a demonstration conference. The rest of the development team, in a meeting at University College London at the occasion, were solemnly listening in with the rewarding feeling of having successively penetrated the maze of so many difficult questions and challenges.

Each of the three sites had an LPC codec attached to a host computer. The three computers communicated through local area nets interconnected through gateways, via Arpanet and Satnet. The packet traffic in that Internet situation (new then!) was a combination of that speech traffic together with "natural" traffic in the Arpanet at the time. Some special knowledge may perhaps be required to fully appreciate the complexity of that experiment. It was one of several major milestones during development of Internet-technology. It took place in 1978 and proved a number of new concepts workable. Intricate logic functioned, and detailed preparation by several collaborating research groups succeeded.

Standardization Procedures

One way to view the Arpanet/Internet effort is development of new technical standards. It even contributed to the way standardization can be done. Traditional telecom standardization takes place in a formal hierarchy of organizations, mostly supported by telecom operating companies. The development of Internet-technology led by ARPA is one prominent example of another form of standards development.

Such international standardization is being driven in a democratic and sometimes bureaucratic environment where participation and progress are dictated as much by political motivation as by technical and commercial interest. Timeliness has sometimes suffered and standards have dropped out of pace with technical and commercial possibilities. Some recent standardization efforts in communication and computers are driven more by directly interested and technically and commercially competent participants. There are many other examples of such technically oriented standards development *forums* today.

At one point in time, late 1980, some of the differences in those two ways of standards development were brought into focus. That was when a meeting in the Packet Switching Protocols Working Group (PSPWG) coincided with a meeting of a group in The International Telegraph and Telephone Consultative Committee (CCITT, now International Telecommunication Union – Telecom Standardization Sector). That group worked on a standard for packet switching X.25. Impressions of the participants were strong. We had something to learn from each other.

Although usually financed by private industry, the success of such ad hoc forums is equally dependent on full openness and access to the results by anybody. Strongly competing actors actively co-operate intimately in development – so that they can go on to compete fiercely for their market shares and livelihood.

There have been examples of large companies setting their own standards and keeping them to themselves. Full openness is now considered necessary for success of standards setting.

From Arpanet to Internet

The network of nets was called Internet.

One fact brought into focus and thoroughly investigated was that various types of traffic have different technical and economic requirements of the transport. Important requirements are error freedom, time delay and economy.

The development in the 1970s resulted in a number of new techniques. Especially important were the protocols TCP and IP (Transport Control Protocol and Internet Protocol). When the "final" standard recommendation was approved and documented internally by the development team, it was based on many years' intensive research and development.

Ten groups carried out that development together as a team. It referred to itself as Packet Switching Protocols Working Group – PSPWG. There were eight groups in the USA, one in England and a small group in Norway. The development comprised investigation of a variety of suggested methods. They were thoroughly studied theoretically and experimentally. The development team presented and discussed intermediate results in daily communication via the new and practical form of communication – electronic mail – and in regular meetings about every three months. A few persons from each group participated – typically 20 to 30 persons in total each time.

The group from ARPA, later called DARPA (Defense Advanced Research Projects Agency), Information Processing Techniques Office – IPTO – led and financed the research and development as a research project of the American Department of Defense as basic technical research.

Resource Sharing Networks

When the research began in 1969 "all" computers were large and expensive. The development had the main purpose to study and develop resource-sharing networks. Resources to be shared were both the "power" of the computers, programs and data of various types – information for shared use. Further, and not least, it was important to create an environment where human resources could co-operate and strengthen creativity and knowledge.

Communicating via Computers

The development of electronic circuit techniques, perhaps the most conspicuous part of computer hardware development, made so-called microprocessors possible from about 1970. Microprocessors are integrated circuits that comprise the main part of a computer. Small chips the size of a fingernail can now comprise entire computers. That development has continued further and further and will probably continue – nobody knows how far. Ultimate limits of complexity have been repeatedly sought, predicted and broken.

Today microprocessors with programs are used as component parts in many devices, more or less specialized.

Among the logical capabilities thus introduced into an apparatus is the capability to execute the rules – protocols – in Internet. That will enable more and more telecommunications to proceed as Internet type traffic. Telecom operators of the world are now eagerly studying the implications. Developing techniques and new types of telecom market demands continue to emerge. We will likely continue to see many great changes of telecommunications and the use of information in the years to come.

Advantage of the Internet

Compared to the traditional telecom network two properties of Internet-technology are foremost: Flexible dynamical use of various carrier media for different demand types and capability to exploit the versatility of computers.

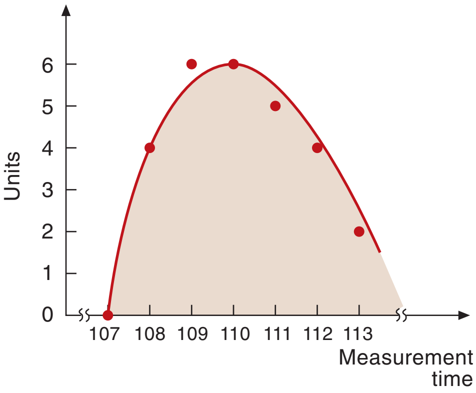
Flexibility concerns several factors. Primarily it has the ability to meet differences in volume of information often expressed as bandwidth of offered traffic – measured as bits per second. Large and small bandwidths may be mixed dynamically, i.e. can be accommodated and optimized while changing rapidly. Secondly there is a great economic potential in exploiting combinations of requirements such as urgency, freedom of error and reliability. Last, but not least, Internet techniques are good at exploiting different carrier media in combination – various cable types, packet radio and satellite net, and probably many new concepts that were less practical without these new techniques.

Some Further Details

National and international telecom networks may be viewed as nodes interconnected by collections of lines of varying properties. Each line may have characteristics such as analog or digital, bandwidth (pulses per second), subject to noise and other factors that determine the capacity and quality of channels. Nodes typically have switches capable of automatically setting up and taking down telephone calls and facilities for manually setting up and taking down other types of channels.

All kinds of information may be represented digitally. Natural values such as temperature, distance, weight, pressure, etc. are measured in specific units and the *number* of units is the digital representation of the value. Accuracy of such representation can be as required. It is a matter of the units and the measurement precision. Depending on the requirements this may be easier said than done of course, but when a measure has been digitized the accuracy of each value is preserved thereafter, unharmed by noise and loss as long as the number symbols can be recognized, recovered and regenerated.

Computers and communications represent numbers in the *binary* system, by strings of the symbols 0 and 1. These are binary digits – bits. Similarly, people represent numbers in the decimal system, by decimal digits – the symbols 0,1,2,3,4,5,6,7,8,9. In a transmission channel or in a storage system symbols are distorted and mixed with noise. In such deteriorated signals it is easier to distinguish between two symbols – than between ten. Therefore, binary representation of numbers is preferred in computers and in digital transmission. Digital channels can be made far more tolerant of noise than analog representation. In practical terms symbols can be restored to perfection, and hence preserve the digitized information unchanged. Binary repre-



Digitizing an analog variable means representing it by measured values at fixed sampling times sentation is easily converted by the computer to and from decimal before being displayed, printed, etc.

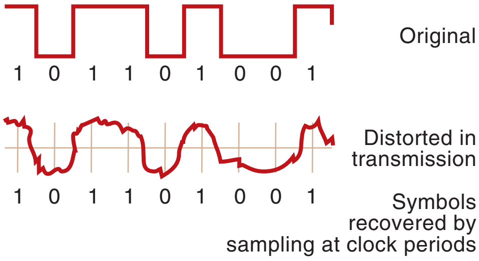
Any kind of information, e.g. speech, music, images, etc. may be digitized. Some types of information such as text and numbers are inherently digital. In particular co-operating computers need to exchange special control information between them. Standards exist that specify precisely how various types of information are represented. Binary information is represented by pulse patterns. Typically, pulses are grouped in "bytes" of eight bits each. Various modulation methods exist for representing bits in transmission channels. Examples are pulse or no pulse, positive or negative pulse, high or low tone, and others.

In its earliest days telecommunications had to live with "the facts of life" that noise and loss along lines were inescapable limitations. Transmission theory and technology have matured over the years. Information carrying capabilities and limitations of various media are now well understood. Techniques have been developed to exploit each medium according to technical and economic criteria.

One way to look at Internet technology from a telecommunications standpoint is that it is a unified method of supporting a large number of different telecom applications together. Varying requirements of individual applications may be specified and met accordingly. Various capabilities of available carrying media may be exploited. These requirements may vary over time and be met automatically in near optimal combination even in dynamically changing situations. Coding methods are available for error detection and -correction to any level of dependability.

Consider the following examples. If a packet comprises representation of monetary amounts, the receiver must not accept any errors inflicted on the packet during transit. Built-in error detection and -correction are paramount. If necessary the damaged packet must be retransmitted until acceptance. And that is more important than time delay. In another case a packet represents a piece of sound, e.g. part of a spoken word. In that case it is more important for the packet to get there in time, perhaps distorted, than to be guaranteed correct and too old to be of any interest.

In a packet switched net the nodes consist of packet switches interconnected by digital channels transmitting streams of packets. The switch has a number of at least three incoming and outgoing channels, two in the case of a gateway between two different types of net. Each incoming packet is transmitted onwards in the right direction according to the address in the packet header and the switch's knowledge of the net. That includes alternative routes and current traffic load. Such knowledge is available to the switch in the form of dynamic routing tables. Queuing, re-packaging and retransmission may happen according to requirements specified in the packet and capabilities of the net, including the current traffic situation.


Original
1 0 1 1 0 1 0 0 1
Distorted in transmission
1 0 1 1 0 1 0 0 1
Symbols recovered by sampling at clock periods

Bit symbols distorted and recovered. Here the symbol "1" is represented by "pulse" and "0" by "no pulse"

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
-	-
-	-
16	1000
-	-

Decimal and binary number symbols

Quality and Efficiency

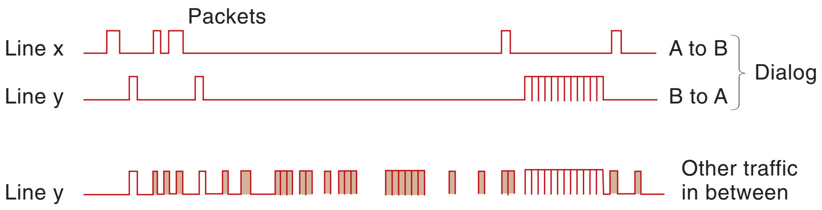
A telecommunications network may be line switched or packet switched and the transmission may be analog or digital. Digital information may be transmitted in analog channels by modem-units that modulate and demodulate to and from analog representation. All these forms of transmission may transfer all types of information. But they have different qualities and efficiencies of different significance for various information types. Analog transmission is traditional for sound and pictures. The information in them is inherently analog. Text and numbers are naturally digital information. Analog transmission has quality limitations, noise and inaccuracy that can only be partly corrected. For normal telephone connections and some others these limitations have little or no practical significance.

Over time digital techniques have been developed extensively and hence should replace older techniques if it were possible to disregard existing assets. Digital transmission is now cheaper and exhibits better quality than analog ones. But several of the existing world-wide telenetworks are still predominantly analog, represent large assets and function well. Further expansion and replacement – analog to digital – imply many questions, trade-offs and techniques for change or coexistence.

As the demand for communication channels between computers and their possibilities increase, digital transmission and also packet switching continue to become more advantageous. That is especially the case for traffic consisting of many and dynamically changing performance factors. An example illustrates that:

Some computer co-operating applications are characterized by burstyness. Machine A sends one or more messages to machine B. Then for some time nothing is sent because A has to do something else, A's user does some thinking, A is awaiting answer or acknowledgement from B, then B may suddenly respond with a large message, and so on. Traffic in the communication channel becomes bursty. Efficiency of such co-operation often requires short transfer time to avoid idle waiting of the receiver. This is typical for answer- and acknowledge-type messages. The length of various messages typically varies a great deal. Transfer time for a message depends on the bandwidth of the channel, i.e. the bit transfer rate. Large bandwidth gets messages through fast. This would tend to make relatively poor use of the channel. It would be idling much of the time. More bursty traffic means less efficient use of the channel because of more idling time. A packet switched channel may be used by a multiplex of packet streams. It may be especially attractive to mix packet streams with different transfer time requirements. Urgent packets slip through fast while packet streams of lower transfer time requirement can wait and make use of otherwise idle periods.

In a packet switched network each leg may carry a mixture of traffic streams belonging to several different "conversations". Each packet is addressed and has necessary identification with it



History of Initial Internet Development

Development Teamwork

The Arpanet collaboration that eventually led to the establishment of the Internet was carried out by a handful of research groups collaborating intimately for many years. Lawrence "Larry" Roberts initially led the Arpanet work as director of ARPA's Information Processing Techniques Office – IPTO. Robert "Bob" Kahn later replaced him. More than anybody else Kahn was the person who formulated goals and guided development of the Internet-technology during the most active development period. Vinton "Vint" Cerf later assisted Kahn. Later on, Cerf promoted and led the further development of the technology and its applications. Vint Cerf today appears as "Internet Guru number one". All these three persons distinguished themselves as excellent professionals and were able to moderate and lead the many capable and outspoken researchers and research groups that took part in the development.

During the 1970s the following ten groups participated in the development of Internet technology as one intimately collaborating team:

- Advanced Research Projects Agency – Information Processing Techniques Office, Washington, DC – *ARPA*
- Bolt Beranek and Newman, Cambridge, Massachusetts – *BBN*
- Stanford Research International, Menlo Park, California – *SRI*
- University of California, Los Angeles – *UCLA*
- Information Sciences Institute, Marina del Rey, California – *ISI*
- Linkabit Corporation, San Diego, California – *Linkabit*
- Comsat Corporation, Gaithersburg, Maryland – *Comsat*
- Massachusetts Institute of Technology, Cambridge, Massachusetts – *MIT*
- University College London, England – *UCL*
- Norwegian Defence Research Establishment, Kjeller, Norway – *NDRE.*

Among these groups the company BBN had many central and decisive roles in the development. BBN was responsible for the daily operation of the network. A Network Control Center – NCC – at BBN supervised the traffic and state of every node in the net. It was important for efficiency of the operation and the experiments that each node (IMP, Interface Message Processor or TIP, Terminal Interface Message Processor) in the entire net could be reprogrammed remotely from NCC. New ideas and variations of protocols were tried all the time. Data from the experiments were recorded and sent via the net to the responsible experimenters at their home locations and to other interested groups. Programs could be tested and errors corrected from the NCC. As the network grew and traffic from connected universities became significant "natural traffic" could be used for experiments in addition to synthetically generated traffic.

Most of the researchers, typically 20 to 30 persons met every third month approximately. The meetings rotated among the sites mentioned above and consisted of detailed discussions following in-depth presentations of results and ideas. The tone was open and could be heated although always friendly. A certain amount of social occasions usually took place and stimulated the smooth co-operative spirit. From day to day the researchers exchanged e-mail. It comprised discussions, experimental results, comments and programs. The assembled group constituted a strong and inspiring research team. From mid 1977 the usual two-day PSPWG (Packet Switching Protocol Working Group) was supplemented by a third – "Internet" meeting dedicated to techniques for internetworking of different nets.

Norwegian Participation

In 1972 ARPA invited the Norwegian Defence Research Establishment – NDRE – to a co-operative effort about so-called resource-sharing networking. Before that ARPA had been in contact with the Norwegian Telecommunications Administration – NTA – with a similar invitation. NDRE then already had long traditions for research co-operation with ARPA. A few researchers at NDRE soon became convinced of the promising prospects of this form of computer networking, and NDRE joined in with ARPA's efforts.

NDRE's work was mainly directed towards packet switched satellite channels. However these channels were integrated with Arpanet at the time and were treated similarly to the leased land-lines. Measurements were made using "Satnet". It consisted of a free channel in a satellite (a digital, 64,000 bits per second channel of type "Spade" in the Intelsat IV satellite) and three ground stations located in Maryland, England and Sweden. The ground station at Tanum, Sweden was owned collectively by Nordic telecom administrations. Some Swedish researchers showed interest in Arpanet, but did not participate in the development. UCLA contributed strongly in the satellite work in addition to the groups at Comsat, UCL and NDRE. It resulted in the satellite channel access protocol called CPODA (Contention Priority Oriented Demand Access). This form of satellite communication has had limited use up to now.

In the early 1970s the Arpanet consisted of both leased lines, mostly of 56,000 bits per second capacity, and of packet radio nets, mainly one in Hawaii and one in the San Francisco Bay area. ARPA now suggested developing packet switched satellite channels as a new information-carrying medium especially for use in resource

PSPWG meetings 1974 – 81. The Packet Switching Protocol Working Group meetings during the most active period of development

10 – 11 Aug 74	On the ferry between Stockholm, Sweden and Åbo, Finland
4 – 5 Sep 75	Linkabit Co, San Diego, California. Host: Irwin Jacobs
12 – 13 Nov 75	UCL, London, England. Host: Peter Kirstein
12 – 14 Feb 76	DCA and ARPA, Washington, DC. Host: Bob Kahn
29 – 30 Apr 76	BBN, Cambridge, Massachusetts. Host: David Walden
29 – 30 Jun 76	NDRE, Kjeller, Norway. Host: Yngvar Lundh
23 – 24 Sep 76	UCLA, Los Angeles, California. Host: Leonard Kleinrock
9 – 10 Dec 76	UCL, London, England. Host: Peter Kirstein
10 – 11 Mar 77	Comsat, Washington, DC. Host: Estil Hoversten
8 – 10 Jun 77	NDRE, Kjeller, Norway. Host: Yngvar Lundh
17 – 19 Aug 77	Linkabit, San Diego, California. Host: Irwin Jacobs
31 Oct – 2 Nov 77	BBN, Cambridge, Massachusetts. Host: Bob Bressler
1 – 3 Feb 78	UCLA, Los Angeles, California. Host: Wesley Chu
3 – 5 May 78	UCL, London, England. Host: Peter Kirstein
31 Jul – 2 Aug 78	MIT Lincoln Lab, Lexington, Massachusetts. Host: James Forgie
1 – 3 Nov 78	Linkabit, San Diego, Caliornia. Host: Estil Hoversten
8 – 11 May 79	BBN, Cambridge, Massachusetts
4 – 7 Feb 80	SRI, Menlo Park, California
14 – 15 May 80	MIT, Cambridge, Massachusetts
7 – 9 Oct 80	UCL, Royal Signals and Radar Establishment, Malvern, England
28 – 30 Jan 81	ISI, Marina del Rey, California

sharing computer networks. They expected, similar to NDRE, that to be of special interest to Norway. NDRE's director Finn Lied, research superintendent Karl Holberg and research engineer Yngvar Lundh were positive to the proposal. The Norwegian participation in the collaboration started in 1972 and was led by Yngvar Lundh. A "Terminal Interface Message Processor" – TIP – was provided by ARPA and installed in 1973. A TIP could connect both to host computers and directly to simple interactive terminals.

ARPA already had a leased line of 9,600 bits per second between Washington and NORSAR, a seismic observatory at the NDRE site at Kjeller, Norway, the result of another earlier collaboration project between (another part of) ARPA and NDRE. That opened a good opportunity for the two divisions of ARPA to co-operate, thus enabling both co-operation in international networking and improvements for the seismic co-operation with Norsar. The 9,600 bits per second line had a multiplexer installed to create two independent channels. The new channel was for use by the Arpanet computer networking experiments. Another line was leased between Kjeller and London where another TIP was installed at UCL. This use of the seismic line thus made it economically feasible to extend the Arpanet to Norway and England.

To build a Norwegian group took some time because of lack of funding. It was hard to convince Norwegian financing sources of the importance of computer networking. For the first two years Lundh's group consisted of two of his graduate students besides himself. In 1975 Paal Spilling, a Ph.D. in nuclear physics looking for a currently more active field of research, was assigned to Lundh's project. Later on, some other well-qualified engineers were assigned, similarly available in NDRE's personnel budget. Persistent invitation by NDRE to NTA's research establishment to participate resulted in the free loan for experimental purposes of a spare channel in the Intelsat IV satellite and a spare line between NDRE and the existing Scandinavian satellite earth station at Tanum, Sweden. This experimental facility including a Satellite Interface Message Processor – SIMP – provided by ARPA was established in mid 1975.

It was the enthusiastic interest of NDRE's research engineers and management for resource sharing networks and new forms of communications that was the decisive factor and driving force of NDRE's participation. Misunderstandings have prevailed in some comments about NORSAR's role in the development. The facts are that NORSAR staff did not participate in the development of Internet technology. The NDRE TIP was placed at NORSAR, which resided in a civilian building just outside NDRE's fence. Hence, access to the TIP was unrestricted, unlike NDRE's buildings which were located on military grounds. Lundh, backed by ARPA, made efforts to create interest in the Arpanet experiments at other establishments, notably the neighbouring large computer centre shared by the University of Oslo and some other academic institutions. During the 1970s such interest was next to non-existent, perhaps due to generally low political esteem of defence related activities in that period of time, despite the basic research nature of this networking research and development. When the NDRE TIP (sometimes referred to as the NORSAR TIP) had been established, interested NORSAR staff began to investigate the possibilities for exchange of seismic data through Arpanet as an alternative to their traditional data exchange connection. As already mentioned they had a leased line for routine data exchange as part of co-operation on seismic research with peer institutions in the US.

Increasing Interest

Most universities, both in Norway and elsewhere, kept well away from the ARPA-collaboration during the 1970s. This situation slowly began to change during the 1980s, because of a change in prevailing political attitudes, or for other reasons. If nothing else, many academic people discovered the convenient communications offered by the Internet. The network gradually became global.

Commercial traffic was prohibited in the Arpanet from the outset and that was still the rule as the network changed into Internet. The network was an experimental facility supported for research purposes. The ban on commercial traffic was lifted in 1991. From then on the number of connected computers and the aggregate traffic began to grow exponentially. In the early 1990s such numbers doubled every seven months, approximately.

From 1994 a few articles in the general press began to mention the Internet as an interesting phenomenon. Since then of course, *Internet* soon flew all around the world as a household word everywhere. From practical obscurity to common knowledge and use world-wide in less than ten years is a remarkable if not unique development in technological history.

Transfer techniques and computer co-operation techniques that are basic to the Internet are rather alien to traditional forms of telecommunication. The established telecom operating companies demonstrated little understanding of Internet technology. That attitude did not change appreciably until the mid 1990s. But from then on telecom operators world-wide have become aware of its great potential and they study it and investigate ways and means for its exploitation on broad fronts.

Visions

Computer- and networking development has led to new applications. Computers can exploit Internet-technology for co-operation. We can expect Internet-technology to be employed for further improved telecommunications. That will no doubt be the case for some telephone traffic, distribution of text, sound and images, including moving pictures, and many new applications and innovations.

Transmission will make use of traditional channels as well as others that we have only thought of in special contexts. That comprises TV cable, local radio nets in many forms and sizes, and local area cable nets of various types including optical fibres.

In short: Internet-technology is capable of meeting many different requirements for traffic types and can exploit many different information-carrying media. The comprehensive development that proposed, analysed and exercised so many possibilities and tested them so thoroughly throughout the whole decade of the 1970s laid the foundation for the rapid growth of applications and its future importance.

Internet Protocol and Transport Protocols

TERJE JENSEN

Terje Jensen (39) is Research Manager at Telenor R&D, Kjeller He earned his PhD degree in 1995 from the Norwegian University of Science and Technology. Activities include performance modelling and analysis, dimensioning and network evolution studies.

terje.jensen1@telenor.com

The Internet Protocol suite has emerged as a pivotal component during the last decade. The basic formats and protocol mechanisms for the protocols related to the Internet Protocol and its common transport protocols are described in this paper.

1 Introduction

The Internet Protocol (IP) has gained a phenomenal place in telecommunications in the latest years. Even the protocol's presence for several decades, certain events and driving forces may well be credited its surf on the promotion wave. The invention of web browsing and openness of the IP and corresponding transport protocols allowing for easy use in education and simplicity of implementation, may be two factors. However, when deploying an IP-based network in a commercial environment, we are faced with several more issues.

The ongoing work related to IP, including protocols, mechanisms, applications, systems, is quite phenomenal. This implies that there is a steady evolution in the area, which is a challenge in keeping track of even the ideas presented within a fairly narrow area. However, in order to follow any discussion going on in different fora, a knowledge of the basic mechanisms and formats of protocols is needed. The objective of this paper is to introduce these formats for IP and the transport protocols.

IP is described in Chapter 2, covering both version 4 and version 6. The IP error and control messages are briefly outlined in Chapter 3. Chapter 4 and Chapter 5 present the two common transport protocols, User Datagram Protocol and Transmission Control Protocol, respectively. Addressing and routing are discussed in Chapter 6.

2 IP Packet Formats

The most fundamental IP service is based on an unreliable, best-effort, connectionless packet delivery system. The service is called *unreliable* because delivery is not guaranteed. The service is called *connectionless* because each packet is treated independently from others, e.g. packets in a sequence may travel along different paths, or some may be lost while others are delivered. The service is called *best-effort* as no packet is assumed to be discarded on purpose.

As explained above, when information is to be passed between two terminals, it is divided into a number of units where each is put into an IP packet (datagram). A network parameter maximum transfer unit (MTU) decides how long fragments can be carried through the network. Commonly, the fragments are reassembled at the destination. The IP packet header formats are treated in this section while issues related to transport protocols are described in the following sections.

2.1 IP version 4

The 1988 version of the IP packet format is depicted in Figure 1. This is also known as IP version 4 (IPv4), where each host has a 32 bit address.

The *Version* field (4 bit) gives the IP protocol version (the current relevant versions are 4 and 6 – note that the format of version 6 is described

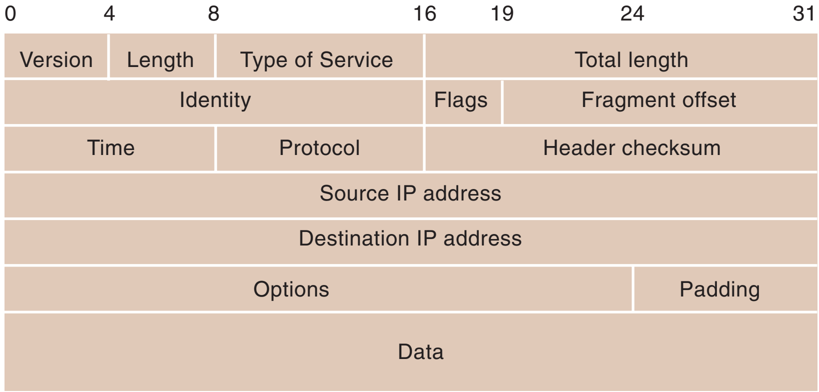


Figure 1 IP version 4 packet format

below). The *Length* field (4 bit) gives the length of the header in number of 32 bits words. Frequently this contains the value 5 indicating 20 bytes (when no options are included).

The 8 bit *Type of Service* (ToS) field gives indications of how the packet should be handled. The original format of this field is given Figure 2.

The three first *Precedence* bits (values 0 to 7) give a kind of priority, allowing a sender to indicate the priority of delivering the packet. When the *D* bit is set, a short delay is requested. When the *T* bit is set, high throughput is requested. Setting the *R* bit indicates that high reliability is requested. In case a router has a number of routes on which a packet can be forwarded, values of the *D*, *T* and *R* bits can be used when choosing which route to use. For instance, if a medium capacity wireline path and a high capacity satellite-based path are available a packet having the *D* bit set could be forwarded on the wireline path while packets having the *T* bit set may be forwarded on the satellite-based path. The interpretation of the ToS field has changed as outlined in Box A.

The field called *Total length* gives the length of the IP packet in number of bytes (including both the header and the data).

The fields in the subsequent 32 bit line are used for fragmentation control. The *Identity* contains an integer that identifies the packet. The intention is to allow the destination to gather all fragments as the *Identity* field specifies which information unit a packet belongs to. The low order 2 bits of the Flags field contain the fragmentation control. The first bit says whether or not the packet can be (further) fragmented. The next bit tells if this is the last packet belonging to an information unit. The *Fragment offset* field gives the offset of the fragment in the packet in the original information unit (given in units of 8 bytes).

The *Time* field specifies how long (e.g. in time ticks) the packet is allowed to remain in the network. In case that time has elapsed, the packet is discarded. This may for instance reduce the amount of packets transported and arriving too late at the destination or being looped in the network. As it is challenging to synchronise all routers, the value in this field could simply be decreased by one for each hop (assuming one unit of time per router). Commonly, seconds is used as time unit, implying that a value of max 255 sec (4 minutes and 15 seconds) can be stated. As each router has to reduce the value with at least one tick, one second per router might then be assumed, which is rather long. This would however depend on the implementation in the routers. Hence, the value is often used as a hop counter.

0	3	4	5	6	7
Precedence	D	T	R	Unused	

Figure 2 Format of the Type of Service field

The *Protocol* field specifies the higher-level protocol (e.g. TCP or UDP). The field called *Header checksum* is used for detecting bit errors in the packet header. The fields *Source IP address* and *Destination IP address* contain the IP addresses (32 bit).

Box A History of ToS

The history of the ToS octet for IPv4 can be found in [ID_ecn]. It goes as follows:

- RFC 791 defined the ToS octet in the IP header. In RFC 791, bits 6 and 7 of the ToS octet are listed as “Reserved for future Use”, and are shown set of zero. The first two fields of the ToS octet were defined as the Precedence and Type of Service (TOS) fields:

0	1	2	3	4	5	6	7	
precedence			TOS			0	0	RFC 791

- RFC 1122 included bits 6 and 7 in the TOS field, though it did not discuss any specific use for those two bits:

0	1	2	3	4	5	6	7	
precedence			TOS					RFC 1122

- The IPv4 ToS octet was redefined in RFC 1349 as follows:

0	1	2	3	4	5	6	7	
precedence			TOS				MBZ	RFC 1349

where bit 6 in the ToS field was defined for “minimise monetary cost”. A motivation for this was the increasing commercialisation of IP-based networks, and some might still ask for “free-of-charge” transfer of packets. In addition to the precedence and Type of Service fields, the last field, MBZ, Must Be Zero, was defined as currently unused. RFC 1349 stated that “the originator of a datagram sets the MBZ field to zero unless participating in an Internet protocol experiment which makes use of that bit.” Furthermore, the 4 bits are considered as values meaning (1000 – minimise delay, 0100 – maximise throughput, 0010 – maximise reliability, 0001 – minimise monetary costs) – other values use default routing/forwarding.

- RFC 1455 defined an experimental standard that used all four bits in the TOS field to request a guaranteed level of link security.
- RFC 1349 is obsolete by “definition of the Differentiated Services field (DS field) in the IPv4 and IPv6 headers”, RFC 2474, in which bits 6 and 7 of the DS field are listed as Currently Unused (CU). The first six bits of the DS field are defined as the Differentiated Service Code Point (DSCP):

0	1	2	3	4	5	6	7	
DSCP						CU		RFC 2474

This shows that assuming a specific interpretation of that octet in the IP header might lead to unintentional actions.

Box B Type Length Value-Formatting

A type-length-value (TLV) format is commonly applied in protocols for each of the fields/attributes. This is applied when a number of optional attributes can be included in a packet. As the name suggests, three fields are then given as depicted in Figure Box B-1.

Type	Length	Value

Figure Box B-1 TLV formatting

The first field gives the type (of optional) attribute, that is identifying the attribute. Then, the next field tells how long the attribute is, e.g. measured in number of octets. The last field contains the value of the attribute itself. Naturally, when only a fixed length is allowed for an attribute, the Length field could be omitted. In case an attribute is mandatory and its position given, the Type field is frequently omitted.

This way of formatting allows for flexibility when including attributes in each of the packets, as well as potential further enhancements by defining new attributes.

A number of options can be included, contained in the *Options* field. These options are for example used for testing and fault detection and localisation. Each option consists of one octet code field, one octet length field and a set of data octets. Options can for example be used for recording the route (each router along the path adds its address), specifying the route a packet should follow (in strict or loose sense) and for recording the time a packet is handled by a router (time stamp inserted). The options are commonly given according to a type-length-value format, see Box B.

The *Padding* field represents bytes containing zeros that may be used to ensure that the packet header is a multiple of 32 bits. The Data field contains the higher level information, like transport protocol and user data.

2.2 IP version 6

In recognition of a growing need for upgrading capabilities of IP, work was initiated to devise a "new" protocol. The major goals of this protocol were to (ref. [Tane96]):

- support more hosts, even with inefficient address space allocation;
- reduce the size of routing tables;
- simplify the protocol, to allow routers to process packets faster;
- provide better security (authentication and privacy) than IPv4;
- pay more attention to type of service, particular for real time data;
- aid multicasting by allowing scopes to be specified;
- make it possible for a host to roam without changing its address;
- allow the protocol to evolve in the future;
- permit the old and new protocols to coexist for years.

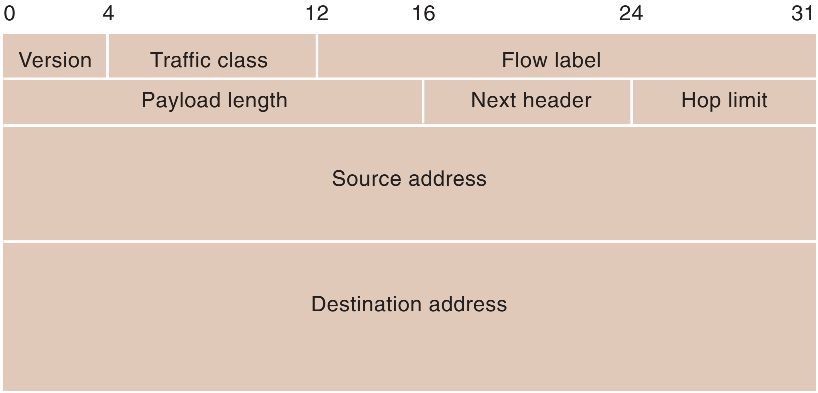
This resulted in version 6 of IP, IPv6, as described in [RFC2460]. The main motivations for specifying IPv6 compared to version 4 were:

- More addresses and addressing capabilities. IPv6 has 128 bit addressing (compared to 32 bit for IPv4). Addressing hierarchy and other grouping of addresses can also be extended.
- Simplified header format. Some of the fields in the IPv4 header have been made optional.
- Better support for extensions. Further options can be introduced.
- Flow labelling. By introducing the flow field, packets belonging to a traffic flow can be explicitly identified, e.g. when they request special handling.
- Improved security capabilities. Extensions are added to support authentication, integrity and confidentiality.

In addition to unicast and multicast addresses, IPv6 also supports anycast. Anycast is like multicast, except that rather than trying to deliver the packet to all destinations specified, it is only delivered to one of them, like the "nearest" one. This could be used for co-operating file servers or any other service where one out of a set of servers may be selected.

Unlike IPv4, only the source may fragment packets when IPv6 is used. In case a router then receives a packet that is too large, it discards the packet and returns an ICMP packet, see Section

Figure 3 Header format of IPv6



3. Another feature with IPv6 is that so-called jumbograms can be supported by using the hop-by-hop extension header (normal packets are limited to 64 kbyte).

The header format of IPv6 is depicted in Figure 3. In addition to the basic format a number of options can be included as described later.

In the same way as for IPv4, the *Version* field gives the version number, i.e. equal to 6 here. The *Traffic class* field indicates traffic class or priorities for packets. It was intended that this field could be used similar to the ToS/DS field for IPv4, see Box A. The 20 bit field called *Flow label* contains an identifier of the packets belonging to the same flow. A flow is said to be a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source may ask special handling by the intermediate nodes. The nature of the special handling can be conveyed by a control protocol (e.g. RSVP or similar) or carried by information within the packets. A flow is uniquely identified by the combination of source address and a non-zero flow label. Hence, assigning a zero flow label to a packet by a source, tells that the packet does not belong to any flow.

The *Payload length* field gives the length of the packet following the basic IPv6 header, in octets. Any extension header fields (options) are considered as part of the payload.

The *Next header* field specifies the protocol header following (e.g. TCP, UDP or any of the header extensions). The *Hop limit* field contains an integer that is decrement by one for each node that forwards the packet. If the value becomes equal to zero, the packet is dropped. This basically replaces the time field in IPv4 realising the way it was implemented by most IPv4 routers anyway. The *Source address* and *Destination address* fields contain the addresses of the sender and the receiver of the packet, respectively, each 128 bits long.

The optional fields are contained in separate headers, where the Next header field specifies which header type that follows. Most extension headers are not processed in the intermediate nodes along a path, only in the source and destination node. One exception is given for the *Hop-by-hop* option header as explained below.

The extension headers should be processed in the same order as they are given. All extension headers begin with a *Next header* field, 8 bits as for the basic IPv6 header. As several of the extension headers may have a variable length, a *Header extension length* field is also given for most of them.

Each extension header should only be included once, except the destination options header that should not be given more than twice; once before the routing header and once before the upper-layer header, see below. In case tunnelling is used, the outer header could be another IPv6 header, again containing its separate set of extension headers. Most of the options within a header follow a TLV-formatting, see Box B, with 8 bits for the type field, 8 bits for the length field and a variable value field.

The following sequence of extension headers is recommended, ref. [RFC2460]:

- IPv6 header as depicted in Figure 3.

- Hop-by-hop options header containing optional information that is to be examined in every node along a packet's path. This is not further elaborated in [RFC2460].

- Destination options header to be processed by the first destination that is given in the IPv6 destination address field and subsequent destinations listed in the routing header. The information in this header is not further detailed in [RFC2460].

- Routing header is used by a sender to list a set of intermediate nodes that a packet has to pass through. This is similar to IPv4 loose source and record route option. The addresses of the intermediate nodes (and the final destination) are given as a list of IPv6 addresses in this extension header. By having a counter that is increased for each node, the node can find which node that is the next one. Then, the address of the next node is inserted in the Destination address field in the IPv6 basic header, see example in Figure 4. Dividing the Header extension length by 2 gives the number of addresses.

- Fragment header is used when a packet larger then the path MTU is to be sent. Unlike for IPv4, the source node is the only one fragmenting packets. The fragment header includes a fragment offset (relative to the start of the original packet) and a unique identifier. When a packet is to be fragmented, all unfragmentable parts of the packet are repeated in all fragments. The unfragmentable parts consist of all fields of the IPv6 packet header and any extension headers up to and including the routing header, if present. Then each of the fragments consists of a repeated unfragmentable part, the fragment header and the fragment itself, see Figure 5.

- Authentication header, see [RFC2402].

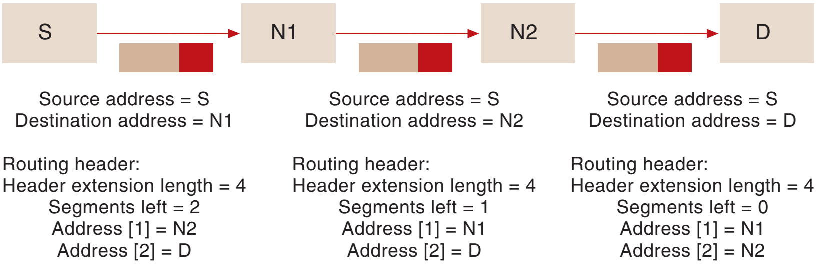


Figure 4 Example of changing the routing header information along a packet's path

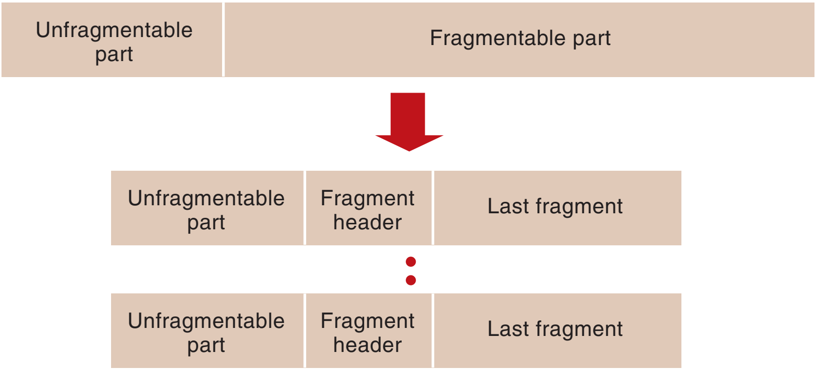


Figure 5 Example of fragmenting IPv6 packets

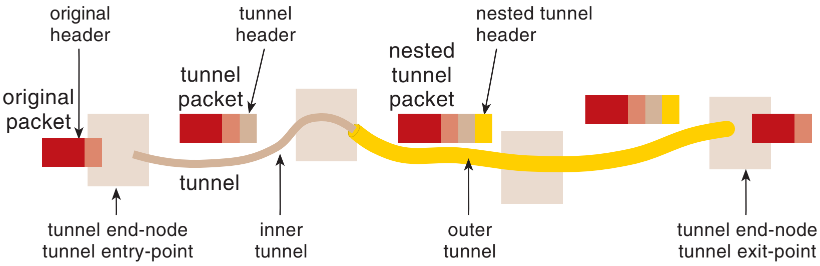


Figure 6 Terms related to IPv6 tunnelling

- Encapsulating security payload header, see [RFC2406].

- Destination options header to be processed only by the final destination.

- Upper-layer header (e.g. TCP, UDP, etc.).

IPv6 requires that all links have an MTU greater than or equal to 1280 octets, although a value of around 1500 octets is recommended. In order to find the largest packets that can be transferred, the path MTU discovery mechanisms as described in [RFC1981] can be implemented.

For various reasons IPv6 can be used for tunnelling other IP packets. One motivation could be to ease the transition from IPv4 to IPv6 by utilising tunnelling during the transition phase. IPv6 tunnelling is a technique of forwarding a packet encapsulated within an IPv6 packet. The forwarding path between the source and the destination of the tunnel packet is called an IPv6 tunnel. For the encapsulated packet, such a tunnel can be seen as a virtual link, as depicted in Figure 6. Tunnels looking like virtual point-to-multipoint links may also be composed.

A tunnel is unidirectional. Two such tunnels can be combined to have a bidirectional tunnelling between the same two end-nodes.

When encapsulating a packet, an IPv6 header (and additional optional extension headers) is prepended to the original packet. This can be nested, that is having several "levels" of tunnels as shown in Figure 6. The source address is the tunnel entry point and the destination address is the tunnel exit point.

Naturally, tunnelling may also be applied for IPv4.

3 IP Error and Control Messages

The Internet Control Message Protocol (ICMP) is seen as a mandatory part of IP. An ICMP message is transferred in the data field of an IP packet. However, ICMP is not considered as a higher-level protocol. This protocol allows hosts/terminals and routers to exchange information for operational and maintenance purposes. Each ICMP message begins with three fields: a one octet type field, a one octet code field that provides further information about the message type, and a two octet checksum field.

In case the ICMP message was initiated as a result of an error occurring when handling an IP packet, the IP packet header of that packet and first 8 octets of the packet causing the error are included. Some uses of ICMP are: testing destination status, reporting on unreachable destinations, flow control (to tell a source to reduce its sending rate), requesting change of routing (e.g. sent by a router detecting an inefficient routing being applied), detecting circular or excessive long routes (e.g. due to the *Time* field in the IP packet header being decreased to zero), detecting incorrect IP packet headers, synchronising clocks and estimating transfer delays (using time stamping functions in the identified router/host), and obtaining a network address and subnet address mask.

IPv6 uses an updated version of the Internet Control Message Protocol, referred to as ICMPv6 (see [RC2463]). ICMPv6 must be implemented by every IPv6 node (being an integral part of IPv6). Similar to above, IPv6 nodes uses ICMPv6 to various tasks, like reporting errors and pinging. Two classes are defined: error messages (destination unreachable, packet too big, time exceeded, parameter problem) and informational messages (echo request, echo reply).

4 User Datagram Protocol – UDP

Avoiding addressing each of the processes utilising IP explicitly, protocol port numbers are introduced as explained in Section 6. Each protocol port is identified by an integer. To communicate with a remote process, the destination IP address and port number of the process have to be known. Then, for a connectionless protocol this information has to be included in every message. Hence, while IP addresses commonly refer to machines/IP layer processes, further identifiers must be used to identify the transport process.

Two protocols dominate in the transport layer; Transmission Control Protocol (TCP) and User Data Protocol (UDP). The former is connection-oriented while the latter is connectionless.

Simplified, UDP can be seen as placing a short header in front of the user data before putting it into an IP packet. Hence, the User Datagram Protocol, UDP, is seen as one of the simplest transport protocols utilising IP. The format of the UDP header is depicted in Figure 7.

As depicted, the UDP header is divided into four 16 bit fields. The *Source* and *Destination ports* identify the UDP processes at the two ends (filling in the Source port is optional). The *Length* field gives the number of bytes in the UDP packet (sum of header and data). The *UDP checksum* field can be used to detect bit errors introduced in the header and user data. A pseudo header is added before the checksum is calculated allowing the receiving process to check that the correct destination and protocol port have been used.

Examining the fields in the UDP header, it is recognised that UDP provides an unreliable connectionless delivery service based on IP. However, UDP adds the ability to multiplex several processes on a given host, see Figure 8.

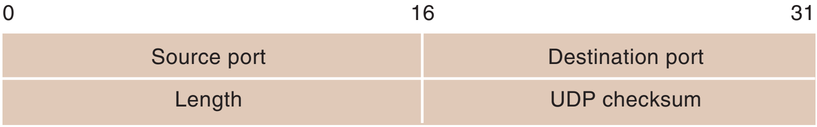
5 Transmission Control Protocol – TCP

Several applications are asking for more reliable transfers than UDP can provide. For those, TCP can be used. TCP provides a reliable transfer service, as packet loss does not need to be dealt with by the application. Two basic features are part of TCP; acknowledgement of transferred data, and window for unacknowledged data units. The former makes sure that the data is correctly transferred by explicitly giving acknowledgements. Introducing windows on the sender side allows a number of packets to be sent before being acknowledged, increasing the utilisation of the network. In contrast to UDP, TCP is connection-oriented. That is, prior to sending data a TCP connection is established between the two end-points.

TCP was specified to support a dependable flow of user data to be carried by IP packets, that is, over an unreliable network. TCP was initially defined in [RFC793], and some corrections and extensions described in [RFC1122] and [RFC1323], respectively. A TCP entity commonly accepts a flow of user data and divides it into packets smaller than 64 kbytes (commonly 1500 bytes are used), and sends each packet to the IP entity. As a network has its maximum transfer unit (MTU), this value is commonly respected by the TCP entity when deciding upon the packet size in order to avoid fragmentation in the IP layer. At the receiver side, the IP packet is transferred to the TCP entity that reconstructs the original user data flow.

TCP uses a sliding window mechanism for efficient transmission and flow control. Sending multiple packets before an acknowledgement has to arrive at the sender increases the transfer rate and therefore the link utilisation. As the window has a limited width, it can also be used to limit the data volume that can be sent and therefore the transfer rate. The window size to use depends on conditions in the network or the receiver (controlled by reporting on window width to be used and delaying acknowledgements). Moreover, as the sender is adjusting the window width based on the packet dropping (lack of timely acknowledgements), dropping packets in the network during congestion will also limit the sender's transmission rate.

Figure 7 UDP header format



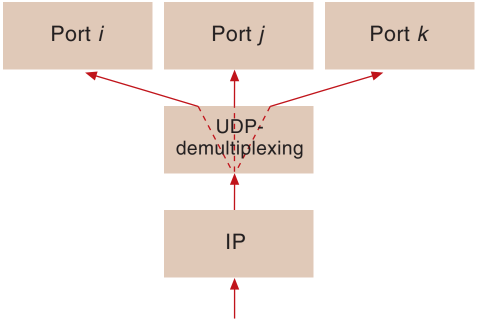


Figure 8 UDP demultiplexing based on port numbers

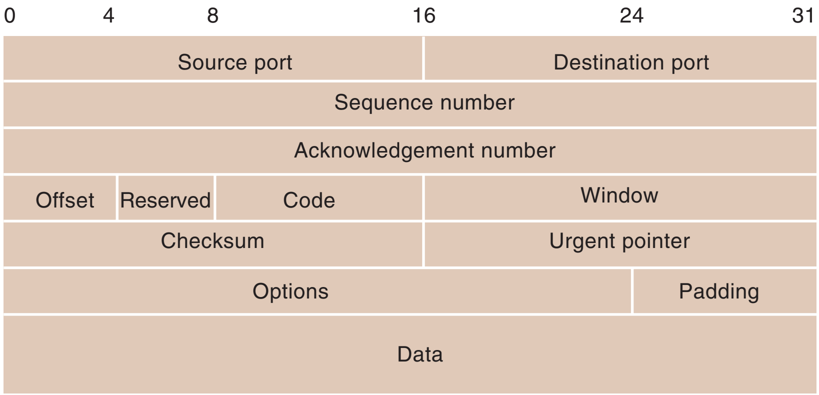


Figure 9 TCP segment format

5.1 TCP Format

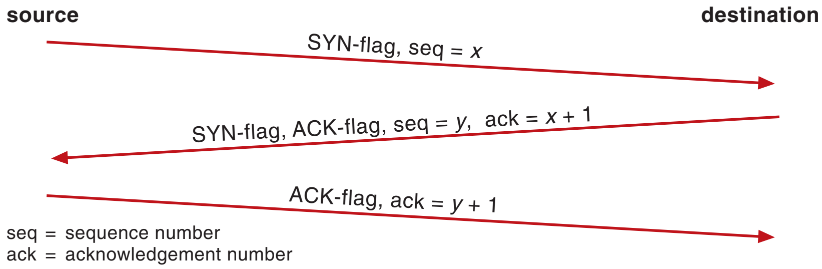
The unit of transfer between two TCP layers is frequently called a segment. Segments are exchanged to establish connections, to transfer user data, to send acknowledgements, to advertise window size and to release connections. The format of a TCP segment is illustrated in Figure 9.

The TCP header starts with the *Source port* and the *Destination port*, similar to the UDP header format. The *Sequence number* identifies the position in the sender's byte stream of the data in the segment. The *Acknowledgement number* identifies the position of the highest byte that the source has received. Note that the sequence number refers to the flow in the same direction as the segment, while the acknowledgement number refers to the stream flowing in the opposite direction as the segment. The *Offset* field contains the integer specifying the offset of the data portion in the segment. This is needed as the length of the Options field varies. After the Offset field, 4 bits are reserved for future use.

The *Code* field gives the contents of the segment by setting 6 flags (urgent pointer is valid – URG, acknowledgement field is valid – ACK, a push is requested – PSH, reset of the connection – RST, synchronise sequence numbers – SYN, sender reached end of its stream – FIN).

The TCP layer at the receiver uses the *Window* field to advertise how much data it is willing to accept. Having the *Urgent pointer* field allows TCP to specify that some data is urgent. The value shows where the urgent data is located.

Figure 10 Messages exchanged for establishing a TCP connection



The *Checksum* is calculated in a similar manner as for UDP by appending a pseudo header.

Note that acknowledgements, windows, etc. refer to volume/byte and not to number of segments. The TCP acknowledgement scheme is called cumulate as it reports how much of the stream has been well received. A motivation for this is that it is simple to implement. In addition, a lost acknowledgement does not necessarily lead to a retransmission. On the other hand, the sender would not receive information on which information that is successfully received and may prepare for retransmitting all segments starting with the one being lost.

5.2 TCP Connection Handling

In order to establish a TCP connection three messages are commonly exchanged as illustrated in Figure 10. In the two first messages the flag called SYN in the Code field is set. In the two last messages the ACK flag as part of the Code field is set.

To close a TCP connection, the FIN flag (part of the Code field) is set. The receiver acknowledges the segment and the session is over.
A TCP connection is unidirectional although acknowledgements are transferred in the opposite direction.

The units of user data seen at the sender side and the receiver side may differ. For instance, at the sender side a TCP entity may receive two blocks of 10 kbyte, while at the receiver side, the TCP entity delivers 20 blocks of 1 kbyte. Furthermore, the TCP entity may collect user data before sending it, unless a *Push flag* is used or *Urgent data* is indicated. Then the information is sent without awaiting further user data.

To accommodate interactive users, TCP provides a push operation that can be used to force delivery of bytes (the PSH flag as part of the Code field is set in the segment). An example of this is when TCP is used to transfer keystrokes.

Similar to UDP, TCP has also some port numbers that are reserved, although most ports are available for dynamic binding. Port numbers below 256 are called *well-known ports* and are reserved for "standard" services, like FTP and Telnet. Having the sender and receiver aligned a pair of sockets sets up a TCP service. Each of the sockets has a socket address consisting of the IP address (32 bit or 128 bit) and a port number (16 bit). A TCP connection can then be identified by the socket addresses at both ends, (receiver_socket number, sender_socket number). A socket can be used for a number of parallel flows. All connections are full duplex and point-to-point.

When a segment has been transmitted by the sender, a timer is started. If this timer expires before the data in that segment is acknowledged, the segment is assumed lost and is to be retransmitted. Which timer value to use could have a major influence on the efficiency of the transfer. Therefore, TCP estimates the round trip time by measuring the time elapsed from sending the segment until the segment is acknowledged. This value can be adapted during the session, as described in the next sections.

5.3 TCP Transmission Policy

The window field in the TCP header is used to announce how much data the sender may transmit on the connection without receiving an acknowledgement. Setting this field by the receiver informs the sender of the volume of data that can be used. For instance, say that a receiver has a 32 kbyte buffer. If the sender transmits a 24 kbyte block, the receiver will acknowledge this block and announce a window of 8 kbyte. If the sender then transmits an 8 kbyte block, it will be acknowledged, but the window size will now be set to 0 (assuming the application in the receiver has not removed data from the buffer). This means that the sender is not allowed to transmit more to this receiver (for this connection) until a higher window size is announced (except for so-called urgent data).

Observe that the sender is not required to transmit the data as soon as it appears in its buffer. Neither is the receiver required to acknowledge the data as soon as the data is received. This is to avoid, for example, situations arising when Telnet is used to transfer keystrokes and responses. In case the keystrokes were to be transferred immediately, much overhead (TCP header and IP header) would be associated with each key that was pushed. *Nagle's algorithm* has been devised addressing this:

> *When data appear at the sender one byte at a time, just send the first byte and buffer all the rest until the outstanding byte is acknowledged. Then send all the buffered characters in one TCP block and start buffering again until they are all acknowledged.*

This is assumed to balance the delay/waiting time observed by the user and protocol/bandwidth efficiency. The algorithm also allows a new packet to be sent if enough data is present to fill half the window of a maximum block. Although often useful for keystrokes, Nagle's algorithm may be less useful for depicting movements of a mouse that are controlled and reflected from a remote host as the mouse cursor could move rather erratically.

Another problem is the so-called *silly window syndrome*. This may occur when data is passed in large blocks, but an interactive application at the receiving side reads the data in small portions (e.g. single byte) at the time. So, when the receiver's buffer is full, a window size of 0 is announced. Then, the application removes a small unit, allowing for the receiver to announce a window of the same small unit. This may immediately trigger the sender to transmit a small unit (with TCP header and IP header), reducing the window size to 0 again. This leads to low efficiency as small packets are transferred. A proposed solution is to prevent the receiver from announcing window sizes of such small units. For instance, the receiver may not send a window update that is less than the maximum block size advertised during connection establishment or its buffer is half full. In addition, the sender may also be restricted from sending small packets, e.g. when the window fits a full block or at least half the receiver's buffer.

Much of the congestion management in today's IP-based networks is placed on capabilities of the TCP. The congestion control applied adjusts the transmission rate. Setting the window size is central for this. The first step, however, is to detect the congestion. For this lost packets are used, e.g. detected by acknowledgements not received before the timeout value has elapsed.

A suitable window size is assigned when a connection is established. For instance, the receiver can specify a window based on its buffer size. When the sender keeps within this window, packets should not be lost at the receiver side, but rather to congestion/errors inside the network. Basically, two potential problems could occur; related to receiver and related to network. Therefore, two windows could be thought of; the window granted by the receiver and a congestion window. Each of these windows gives the number of bytes that the sender may transfer. The effective number of bytes that can be transmitted before being acknowledged is the minimum of the two window sizes. When a connection is established the sender's congestion window is initialised to the size of the maximum segment that can be used. It then sends one maximum segment. If this segment is acknowledged before timeout, the sender adds one segment (counted in number of bytes) to the congestion window, making it two maximum segment sizes. Then two maximum segments can be sent. For each of these segments acknowledged, the congestion window is further increased by one segment size. Therefore, when the congestion window is *k* segments, if all *k* are acknowledged on time, the congestion window is increased by *k* segment sizes, that is, effectively doubled. This is

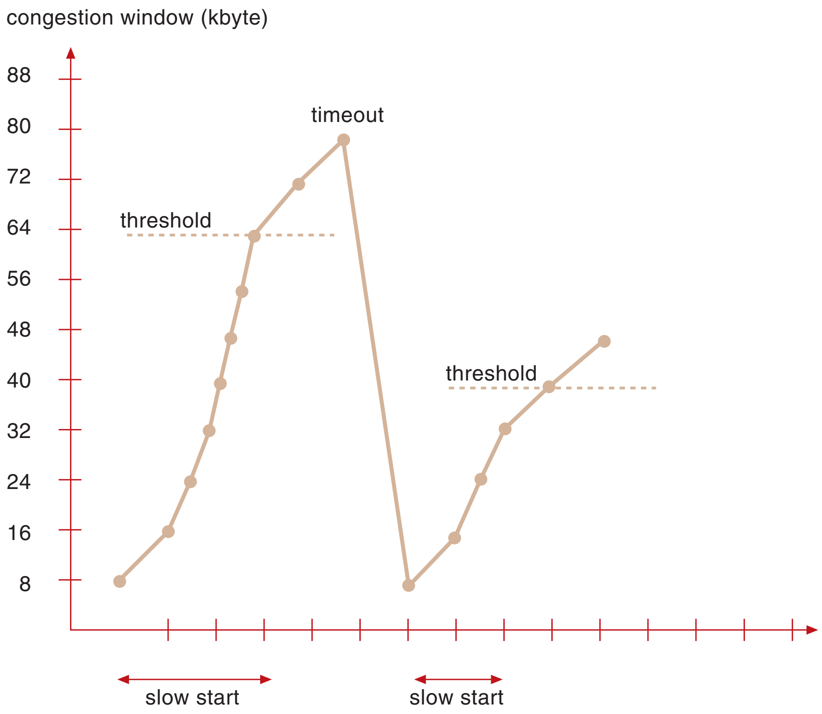


Figure 11 Illustration of the congestion algorithm; slow start, thresholds (maximum segment size assumed to be 8 kbyte)

illustrated in Figure 11 (one maximum segment size is assumed to be equal to 8 kbyte). The congestion window continues to grow until a timeout occurs or the receiver's window is reached. This part of the algorithm is called *slow start*.

A parameter called *threshold* is also used. This is initially set to 64 kbytes. When a timeout occurs, the threshold is reduced to half of the current congestion window. That is, assuming that a timeout occurs when the congestion window is 80 kbyte (as in Figure 11), the next threshold will be 40 kbyte. Moreover, the congestion window is set to one segment size (equal to 8 kbyte in Figure 11). So after a timeout, the slow start approach is again applied until the threshold is reached.

When the congestion window is greater or equal to the current threshold, its size is increased by one segment size (i.e. in a linear manner) as shown in Figure 11.

The two algorithms described above, slow start and congestion avoidance, are essential parts of TCP. Slow start and congestion avoidance are independent algorithms with different objectives. When congestion occurs TCP slows down its transfer rate of packets. Then the slow start algorithm is invoked again. These algorithms require that two variables are maintained for each TCP connection, a congestion window, *cwnd*, and a slow start threshold size, *ssthresh*. Then, these operates as follows (ref. [RFC 2001]):

1. Initialise *cwnd* to one segment and *ssthresh* to 65535 bytes.

2. The TCP output routine never sends more than the minimum of *cwnd* and the receiver's announced window.

3. When congestion occurs (timeout or duplicate acknowledgements), one-half of the current window size is saved in *ssthresh*. Furthermore, when a timeout occurs, *cwnd* is set to one segment.

4. When new data is acknowledged, *cwnd* is increased as: i) in slow start (*cwnd* is less or equal to *ssthrresh*) increase by one segment – in effect a doubling of the window for each round-trip time; ii) in congestion avoidance increase by *segsize*segsize/cwnd* (*segsize* is the segment size), which is a linear growth.

5.4 TCP Timers

TCP works with several timers, at least conceptually. The *retransmission timer* is started when a segment is sent. If an acknowledgement is received before the timer expires, the timer is stopped. On the other hand, if the timer expires before the segment is acknowledged, the segment is retransmitted and the timer is restarted. A main question is how to set a proper value of this timer. Setting it too short will result in many unnecessary retransmitted packets. Setting it too long will result in long retransmission delays and possible low throughput. Therefore, it is desirable to have an algorithm that dynamically adjusts the timer value based on measurements of the round-trip delay. A variable, *round-trip time (RTT)* is maintained by TCP, which is an estimate of the round-trip delay. Whenever an acknowledgement arrives, the TCP entity measures how long the acknowledgement took. It then updates *RTT*, for the next interval $i + 1$, using a smoothing factor, say a:

$$RTT_{i+1} = a \cdot RTT_i + (1 - a) \cdot T$$

where T is the last measured round-trip time. Typically (ref. [Tane96]), a is 7/8. Then, TCP commonly uses a value $b \cdot RTT$ as the value of the retransmission timer. One proposal is to set b in proportion to the deviation of the round-trip time. However, to simplify the calculations the deviation, D, can be updated as:

$$D_{i+1} = c \cdot D_i + (1 - c) \cdot |RTT - T|$$

where c is a smoothing factor. A commonly used (ref. [Tane96]) value of the timeout is then $RTT + 4 \cdot D$.

When a timeout occurs, one has to decide how to update the variables. According to the so-called

Karn's algorithm, a feasible approach is not to update *RTT* on segments that have been re-transmitted, but instead double the retransmission timer until the segments get through the first time.

As mentioned above, TCP uses a retransmission timer to ensure data delivery upon missing ACK messages. The duration of this timer is referred to as the *retransmission timeout (RTO)*. A basic algorithm for computing the *RTO* is described in [RFC2988]. A TCP sender maintains two state variables, *smoothed round-trip time (SRTT)* and *round-trip time variation (RTTVAR)*. A clock granularity of *G* is assumed. Then, the following five rules are to be obeyed:

i) Until a round-trim time (*RTT*) measurement has been made *RTO* should be set to a value between (2.5 + *G*) seconds and 3 seconds.

ii) When the first *RTT* measurement, say *R*, has been made: $SRTT = R$, $RTTVAR = R/2$, $RTO = SRTT + \max(G, 4 \cdot RTTVAR)$.

iii) When a subsequent *RTT* measurement, say *S*, has been made: $RTTVAR = (1 - b) \cdot RTTVAR + b \cdot |SRTT - S|$, $SRTT = (1 - a) \cdot SRTT + a \cdot S$ (in the given sequence), where $a = 1/8$, and $b = 1/4$. Then $RTO = SRTT + \max(G, 4 \cdot RTTVAR)$.

iv) Whenever *RTO* is computed, if a value less than 1 second is obtained, *RTO* should be rounded up to 1 second.

v) A maximum value of at least 60 seconds may be assigned to *RTO*.

Furthermore it is stated that Karn's algorithm has to be used when taking *RTT* samples, meaning that retransmitted segments must not be considered unless the timestamp option of TCP is applied. At least one *RTT* measurement per *RTT* has to be taken (unless Karn's algorithm prohibits it).

When the retransmission timer expires, the earliest TCP segment not acknowledged is retransmitted and $RTO = RTO \cdot 2$. A maximum limit may be used, as stated above. For some TCP implementations, *SRTT* and *RTVVAR* may be cleared when a segment is retransmitted several times (and *RTO* has been doubled several times). Then, when a proper *RTT* estimate is found these variables are initialised again according to ii) above.

Another timer is the *persistence timer*. This is used to avoid deadlocks that can occur when the window size is set to 0. After receiving this window size, the sender initialises the persistence timer. If this timer expires, a probe packet is sent to the receiver, who replies with the window size. If the size is still zero, the persistence timer is set again.

The *keep alive timer* is sometimes used when a connection is idle for a long time. This timer is used to check if the other side is still there. If a reply is not received from the other side, the connection is released.

The last timer present in several TCP implementations is used during connection release to make sure that all packets belonging to a connection have arrived (or been lost).

5.5 TCP Friendly Rate Control

In a best-effort IP-based network, supporting streaming services may ask for particular concerns. Such a concern is to limit the variation in the throughput. A protocol variant to address this is presented in [ID_tfrc], called *TCP Friendly Rate Control (TFRC)*. A drawback of having smoother throughput is that changes in available bandwidth are responded to more slowly. Hence, TFRC uses a throughput equation when calculating the sending rate also considering the loss ratio and round-trip time as for ordinary TCP. The equation for calculating the throughput, *X*, as given in [ID_tfrc] is:

$$X = \frac{s}{R \cdot \sqrt{\frac{2p}{3}} + T \cdot 3 \cdot \sqrt{\frac{2p}{8}} \cdot p \cdot \left(1 + 32p^2\right)}$$

where

s is the packet size in bytes (some streaming applications have fixed packet sizes, otherwise an average measure might be applied);

R is the round-trip time in seconds;

p is the loss ratio;

T is the TCP retransmission timeout value in seconds.

A further simplification is suggested by setting $T = 4R$ (or $T = \max(4R, 1 \text{ second})$). An argument for the equation given is that it should give fairly similar values to when the TCP rate calculation function is applied. Basically, the transfer rate is doubled or halved when an acknowledgement is received or missed, respectively. However, modifiers are used to limit the rate variations as described in [ID_tfrc].

5.6 High-capacity Links and TCP

A commonly referred quantity when discussing performance is the *bandwidth-delay product*. It is obtained by multiplying the bandwidth by

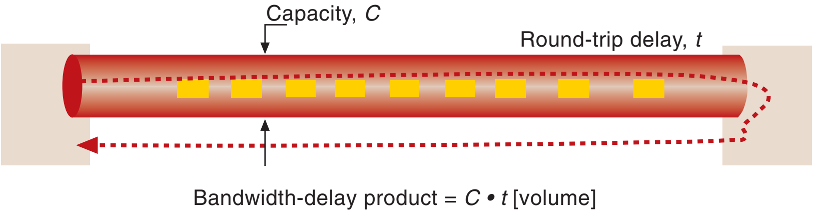


Figure 12 Bandwidth-delay product

the round-trip delay. An observation is that the receiver's window should be at least as great as this product. If this window is too small, the sender has to pause in the transmission waiting for the data to be acknowledged. This could result in poor utilisation of the transmission link. On the other hand there is rarely only one single sender-receiver connection present at any major link.

As transmission rates increase, the bandwidth-delay product may very well increase for an end-to-end connection. This may ask for additional features in TCP in order to efficiently utilise high rate links. Due to the window mechanism TCP performance does not only depend on the transfer rate but also on the round-trip delay. The bandwidth-delay product gives a measure of the amount of data that can "fill the transfer link" see Figure 12. Note that the bandwidth may not be symmetric. Additional TCP performance challenges arise as this product gets larger.

Three fundamental problems are, ref. [RFC1323]:

- Limit of the window size: Using 16 bits to specify the window size limits the largest size to 64 kbyte. This can be circumvented by introducing an additional TCP option, called window scale. Then the TCP window can be interpreted as a 32 bit value by considering the scaling factor that is given in the window scale field of a TCP SYN segment. This means that the scaling is fixed in each direction at the establishment of the connection. When scaling the window, the value in the field is right-shifted a number of positions according to the value in the window scale field (binary shift). This is the same as saying that the scale factor is given as a power of two and coded logarithmically. Say scaling by $8 = 2^3$ means sending the value 3.

- Loss recovery: As more data could be under way for a large bandwidth-delay product, a dropped packet may have large impact on the resulting throughput and more data might be scheduled for retransmission. Selective acknowledging packets and selective retransmission could alleviate this.

- Estimating round-trip delay: Having an accurate estimate of the round-trip delay is essential to avoid unnecessary retransmission at the same time as dropped packets are retransmitted without delaying the sender too much. Therefore, the *retransmission timeout interval (RTO)* has to be set properly based on estimates of the *round-trip time (RTT)*. An additional TCP option including a timestamp allows for improved estimates of *RTT*. Then the sender assigns a timestamp to the packet and the receiver returns this timestamp in the ACK segment (timestamp echoing). This implies that the timestamps returned are related to the ACKs that advance the window avoiding the introduction of "artificial delays" due to sequence buffering in the receiver (when ACKs are delayed and several segments are to be acknowledged, this rule is not followed).

Two further mechanisms are described in [RFC1323]; *Round Trip Time Measurement* (RTTM) and *Protect Against Wrapped Sequences* (PAWS). The former addresses estimating round-trip delays, basically by introducing time stamps in the segments.

When large amounts of packets can be on the way, it may happen that sequence numbers are to be used several times for the same connection within a short time. This could cause sequence numbers to be wrapped-around and also that "old" packets arrive after the retransmitted ones, possibly being mistaken for a packet belonging to a following round of sequence numbering. The PAWS mechanism has been proposed to avoid this. The same mechanism as RTTM is used, i.e. timestamps inserted into the segments. Basically, if a segment is received for which the timestamp is too old, this segment is dropped. What is considered as too old is found by comparing the segment's timestamp with the timestamp of the recent segment updating the counter for the RTTM mechanism.

5.7 TCP and Short Transactions

For many applications a small amount of information is to be exchanged by the end-points. An example may be a message containing status information of a network element reported to a management system. Having to establish and release (as well as maintain) a TCP connection for such exchanges would imply additional overhead in terms of messages and delays.

An extension to TCP for supporting (short) transactions more efficiently is described in [RFC1644]. Transactions are commonly used for the client-server oriented end-user applications as well as for several control and management procedures. The objective of the TCP extension

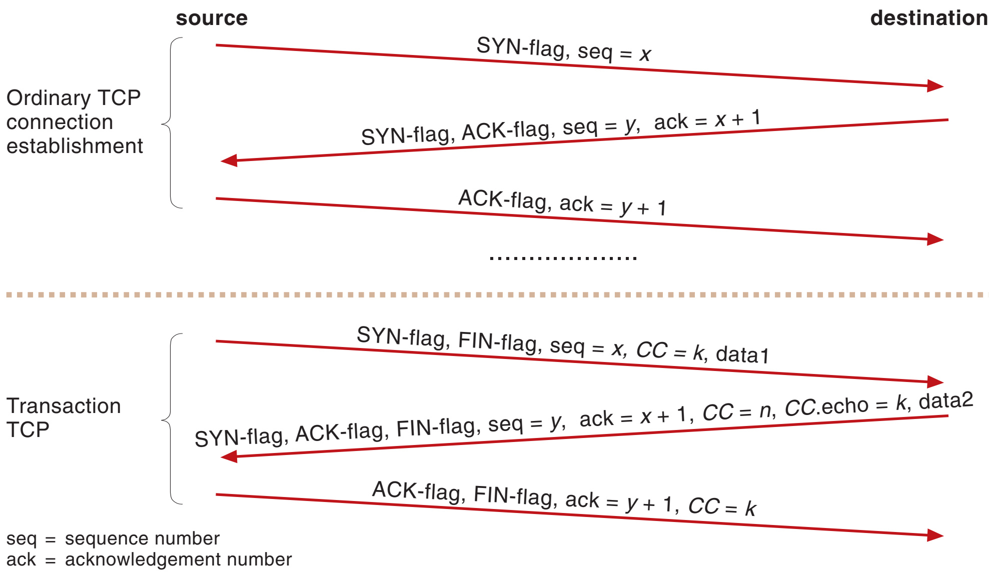

source
destination
Ordinary TCP connection establishment
SYN-flag, seq = x
SYN-flag, ACK-flag, seq = y, ack = x + 1
ACK-flag, ack = y + 1
....
Transaction TCP
SYN-flag, FIN-flag, seq = x, CC = k, data1
SYN-flag, ACK-flag, FIN-flag, seq = y, ack = x + 1, CC = n, CC.echo = k, data2
ACK-flag, FIN-flag, ack = y + 1, CC = k
seq = sequence number
ack = acknowledgement number

Figure 13 Message sequence for establishing an ordinary TCP connection and for the complete transaction/TCP transfer

is to complete the transaction by exchanging a few segments in each direction, thereby reducing overhead by explicit connection establishment and release.

A 32 bit incarnation number is introduced (called *Connection Count, CC*). This is introduced as a TCP option. The CC values are increased for successive connections. The receiver of a segment can check against its list whether or not the received segment is a new one. Thus, the receiver has to keep a list of the latest used CC values for all clients (for a while, e.g. given by twice the *Maximum Segment Lifetime, MSL*). The CC number is also echoed in the return segment in order for the sender to correlate the response with the original request.

An ordinary TCP establishment sequence would then not be needed, see Figure 13. A minimal sequence consists of three segments; firstly a request – SYN segment (say $CC = k$); secondly a reply – SYN segment (say $CC = n$, CC.echo $= k$); and thirdly an acknowledgement segment. In case the second segment is not a segment containing the corresponding flags set, an ordinary TCP establishment and sequence is entered. When a transaction consists of three segments only, estimating the *RTT* could be based on several transactions.

5.8 TCP on Asymmetrical Configurations

When there is an asymmetrical configuration for the transfer rate, TCP may face additional challenges. This is particularly the case when the path from the receiver to the sender (reverse direction) for sending TCP ACK segments has a significantly lower rate than the other way (forward direction). Both low rate and short buffers in the reverse direction may lead to TCP ACKs not arriving at the sender in time for not restricting the sending rate. Two key issues have to be addressed: i) manage bandwidth usage on the reverse link, e.g. to limit the number and transmission capacity for transferring ACKs; and ii) avoid any adverse impact of infrequent ACKs. Some approaches to deal with the former are, ref. [ID_pilc]:

- TCP header compression; reducing the size of the header.
- ACK filtering; drop ACKs that may not be needed, e.g. by dropping ACKs in the buffer when a new ACK arrives.
- ACK congestion control; having a mechanism that indicates to the receiver that the ACK path is congested and the receiver's response to such an indication, e.g. using Random Early Detection (RED) and Explicit Congestion Notification (ECN), see [Jens01].
- ACKs first scheduling; placing ACKs first in the buffer.
- Back pressure and fair scheduling; limiting the amount of data packets in the reverse direction.

Commonly, several of these techniques have to be combined.

Receiving ACKs infrequently might lead to the sender halting its transmission. This can be tackled either end-to-end or locally at the constrained link. Some approaches to handle infrequent ACKs are, ref. [ID_pilc]:

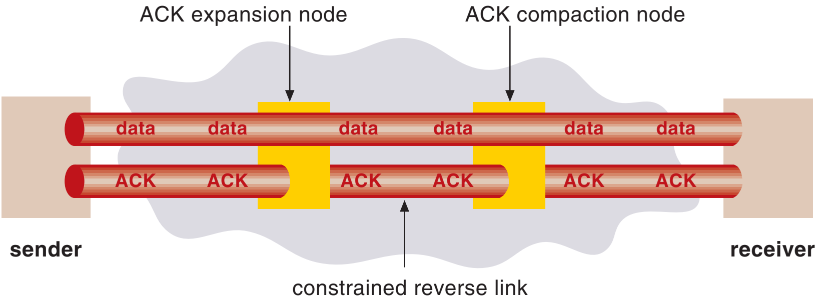


Figure 14 Introducing ACK compaction and expansion nodes (note these might be the receiver and the sender nodes, respectively)

- Sender adaptation; when an ACK message acknowledges a number of data segments, one could equate it to the same number of ACK messages. Thus, the congestion window at the sender side can grow correspondingly. Making sure that the number of segments acknowledged and not the number of ACK messages is used would also be in line with the situation on the forward direction.

- Reconstructing ACK messages; in case sender adaptation is not used, the original sender side of the constrained link (forward direction) can examine the ACK messages and reconstruct any intermediate ACK messages not seen. This could be done by generating the ACK messages and putting them on the link evenly distributed in time.

A more generic technique of the latter is to introduce an ACK compaction and an ACK expansion in the network, see Figure 14. The compaction would remove any “unnecessary” ACKs and the expansion would reconstruct the same ACKs, making it transparent for the end-systems. However, additional protocol mechanisms have to be introduced to enable this.

6 Addressing and Routing

6.1 Addressing and Identifiers

A *name* identifies *what* an object is, an *address* identifies *where* it is, a *route* tells *how* to get there, a *path* says which sequence of *steps* to traverse, and a *link* would be a *step* in the path.

Referring to the configuration depicted in Figure 15 a major issue is how the different port numbers are allocated. In principle there are two ways this can be done; i) universal assignment; and ii) dynamic binding. In the former, a centralised authority is typically used to assign port numbers and publish the results to the hosts (could well be done hierarchically). In the latter, port numbers are assigned when needed. This implies that each program that needs a port number is assigned one on demand. In order to know the port number on a remote host, an enquiry has to be sent to that host, which replies with the proper port number to use.

Typically a combination of the two ways of assigning port numbers has been chosen; some numbers are fixed while others are used dynamically. The ports refer to identifiers used on the transport layer (TCP/UDP). The port identifiers are included in the UDP/TCP headers.

Referring to identifiers used on the IP layer, a Domain Name System (DNS) is used to translate between more high-level names and IP addresses. For instance, the name *viking.telenor.com* could translate into a 32 bit IP version 4 address (and the other way around). Such a naming scheme would then be used to assign names throughout the IP-based networks. It also provides a large-scale example of the client-server concept as a DNS server would be enquired in order to make a translation between the name and the address, see examples in Figure 16.

In principle, the resolution algorithm used for the translation proceeds from the top (top-level domain) and continuing down. There are two ways of using the domain name system: i) by asking the name servers one by one until the resolution is complete; or ii) by asking a name server to do the complete resolution (lower

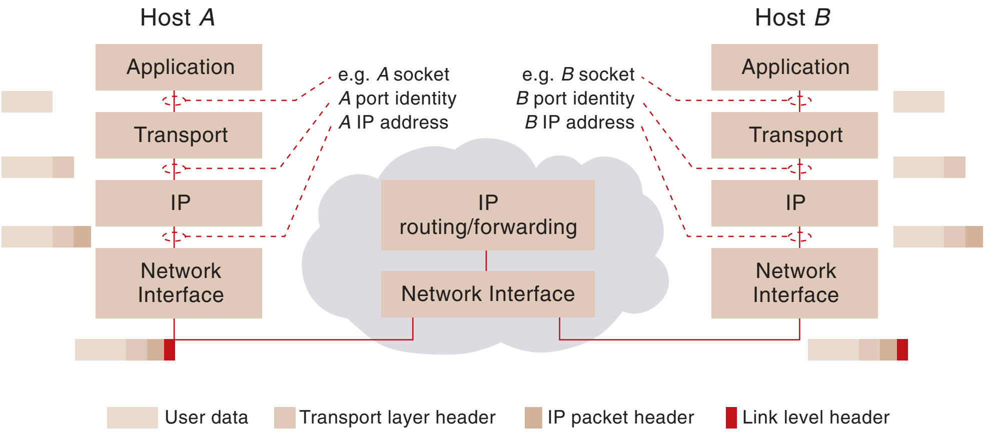


Figure 15 Hierarchy of functionality/protocols

option in Figure 16). In both cases, the requester forms a query that includes the name to be resolved, in addition to other fields. When a server receives a request, it checks to see if the name is within its area (in the data base). If so, it translates the name into the address and appends this result to the request before returning the reply message. If the server is not able to resolve the name, it forwards the request to the next name server (if a complete resolution was indicated in the request) and returns the result to the requester, or replies its lack of information to the requester (if no complete resolution was indicated). To initiate this procedure, all machines must know the address of at least one domain name server.

In order to have more efficient name resolutions (as well as to increase the availability of the resolution function), local name servers are commonly used. Such a local server may keep a list of all names recently being resolved (recently refers to a time less than a time-to-live which could be set per name). Asking the local server first would frequently make the resolution function more efficient as it turns out that more requests are initiated for the same domain name. Resolving a domain name is also commonly referred to as name binding.

As seen from the example, the names/addresses are hierarchical. The last part of the name (after the last period) tells which "area" out of the first/top-level separation. For the IP-address the first

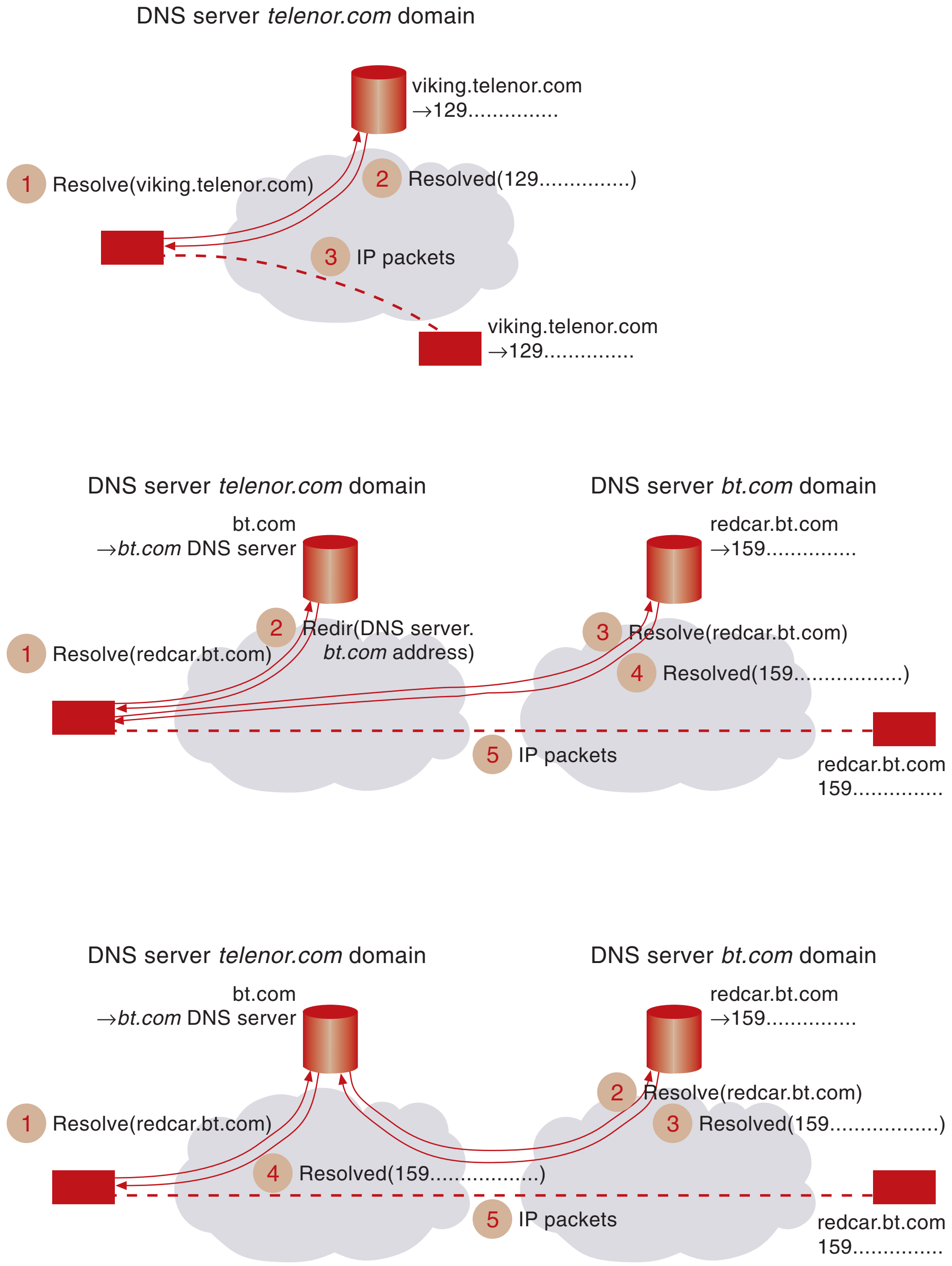
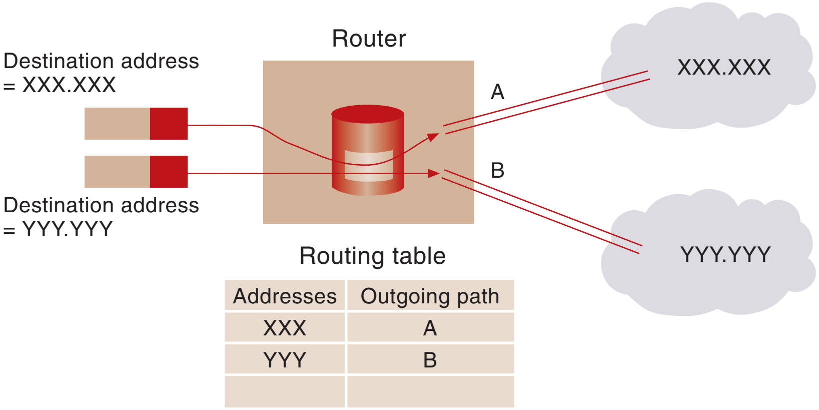


Figure 16 Examples of sequences for name resolution. Note that messages and addresses are fictive



Addresses	Outgoing path
XXX	A
YYY	B

Figure 17 Illustration of routing information in a router

part (before the first period) has a similar meaning. However, more "fields" are commonly needed to identify precisely the "area". An organisation is assigned the responsibility to administer the name range at a certain level. For example, Telenor would by itself assign names within the *telenor.com* range. Examples of some top-level domain names are *.com*, *.edu*, *.gov* and *.org*. Recently, others have also been accepted (*.biz*, *.pro*, *.info*, *.name*, *.aero*, *.museum* and *.coop*). In addition, most countries have their own domain name. The domain names may not contain information about the physical location of a host/machine. This is one reason for the need to translate the name into an IP-address.

Inverse translation from IP address to name could also be asked for. In particular, this is often used for names written in so-called dotted-decimal form; e.g. *abc.def.ghi.klm*, where all these are digits in the range [0...9].

6.2 Routing

During the initial phase of the Internet (ARPANET) the names and addresses of all computers attached were kept in a single file that was edited by hand and then distributed to every site. By the mid-1980s it was clear that such an approach would not suffice any more. This goes for both the capacity to update the information (the single file) and the capacity to distribute the file to every relevant location.

As mentioned above distinctions are made between names, addresses and routes; a name indicates what one seeks; an address indicates where it is; a route indicates how to get there. The IP packets deal primarily with the addresses, while higher-level protocols may take care of the mapping from names to addresses. The mapping from address to route is carried out in each of the routers examining the IP packet header. In the following sections, a brief overview of routing is given. Some more details are presented in [Feng01].

6.2.1 Routing Algorithms

The routing algorithm can be considered as the part of the network layer responsible of deciding which output line an incoming packet should be forwarded on. For a connectionless service, the routing has to be done for each packet, while for a connection-oriented service, routing is exercised on the establishment of the connection. The latter may be called session routing as the routing decision remains in force for the session.

The term routing refers to the process of selecting a path along which packets are sent. Conceptually, one may think of the routing table in a router as consisting of pairs; the set of addresses, and the outgoing path to use (as seen for that router). This is schematically illustrated in Figure 17. Observe that the complete destination address does not need to be specified in the routing table.

A distinction between routing and forwarding is also essential. The latter refers to the process of transferring an IP packet to an outgoing link when it arrives. Hence, the routing procedure finds which information to insert into the routing table, while forwarding sends packets on the next hop according to the routing table data.

Executing routing algorithms may require involved calculations. Therefore separating these processes and running them on different processors may allow for higher forwarding throughput.

Basic properties requested from a routing algorithm are: correctness, simplicity, robustness, stability, fairness and optimality [Tane96]. Two major classes can be identified for routing algorithms:

- Non-adaptive algorithms that do not use measurements or estimates of current traffic load or topology in the routing decisions. These are also called static routing algorithms.

- Adaptive algorithms that change their routing decisions depending on changes in topology, perhaps also the traffic load. These may further differ in the way the information is distributed, how frequently the routing is changed and which metrics are used.

A few groups of routing algorithms are e.g. [Tane96]:

- Flooding; every incoming packet is sent on every outgoing link except the one it arrived on. One may avoid too many packets by using a packet hop counter, dropping packets which have already visited the node (by keeping track of which packets have already been there), or selective flooding (sending on outgoing links going approximately in the requested direction).

- Flow-based routing; taking both traffic load and topology into account when deciding the routing. In case the routing is static, mean traffic load values could be used (e.g. found from measurements).

- Distance vector routing; each router maintains a table (vector) describing the better "distance" to each destination (or rather set of destinations) and the corresponding outgoing link. As these tables are updated by information exchanged between routers, it can dynamically adapt to the current situation in a network. The "distance" measure can be number of hops, delay, queuing length, and so forth. A well-known problem is the count-to-infinity problem when a node goes down as this information may propagate slowly.

- Link state routing; the topology and all delays are estimated and distributed to every router in the network. Then, a shortest path algorithm can be used to find which path to use to reach each (set of) router. Four steps can be recognised: i) discover neighbour routers and their addresses (e.g. by Hello packets); ii) measure delay (or cost) to each of the neighbours (e.g. by Echo packets); iii) distribute a packet to all other routers containing the measure; and iv) compute the shortest path to every (set of) router.

- Hierarchical routing; routers may be divided into a number of regions. All routers know every other router within its region although without knowing details of other regions. Information on which router to use in case a packet is to leave the region has to be known, though. Several levels may be used in the hierarchy depending on the number of routers, as well as other factors, like domains, operators, etc.

- Broadcast/multicast routing; sending packets to multiple receivers is commonly seen for some services, like news and conferences. One simple way of broadcasting is simply to distribute an incoming packet on all outgoing links (flooding), although this would likely waste transmission capacity. Then, reverse path forwarding has been proposed where a router forwards a packet from a sender in case the packet arrives on a link that is used for sending packets to the sender (in the opposite direction). Spanning trees can also be constructed when packets are to arrive at a set of destinations. However, maintaining such trees may become cumbersome for larger networks.

  Therefore a core-based tree algorithm has been suggested where a smaller core does the multicasting within the core and then other multicast techniques can be applied between the core and the final destination. This may be seen to have some resemblance with hierarchical routing.

Link state routing is commonly applied within a domain, e.g. Open Shortest Path First, OSPF and Intermediate System – Intermediate System, IS-IS. Commonly topology information is used (only able to tell whether links are up or down). More dynamics can be reached by introducing other measures. To include a measure based on delay may only cause oscillations to occur as traffic tends to be sent towards paths with short delays. These paths may then become overloaded and long delays result of which other links will be announced as lighter loaded and then the traffic may be sent towards these, and so on.

Therefore, other measures and constraints are also considered, ref. [Feng01]. When several links are included in a path, the measures of the links have to be aggregated. Aggregating metrics depends on the parameter as described in Box C. Finding an optimal path subject to two or more additive, multiplicative or root-mean-square is NP-complete (cannot be solved in polynomial time). Hence, heuristic algorithms are used with such measures.

In addition to specifying the next router, looser routing can be applied. Then a set of routers is listed, allowing for flexibility to decide which one to use. Further, this allows a sender to have imperfect information on the details. The set of routers may also be referred to as an abstract router. This is support source specific routing when the complete path is, more or less, strictly specified by the sender.

Box C Aggregating Measures

When aggregating measures a number of different approaches may be feasible, depending on the parameter in question. Assuming independence between the different parts, three ways are:

- additive basis, i.e. $P_{tot} = P_1 + P_2 + \ldots P_n$
- probabilistic basis, i.e. $P_{tot} = 1 - [(1 - P_1) * (1 - P_2) * \ldots * (1 - P_n)]$
- root-mean-square basis, i.e. $P_{tot} = \mathrm{sqrt}[P_1^2 + P_2^2 + \ldots P_n^2]$
- concave basis, i.e. $P_{tot} = \min[P_1, P_2 , \ldots, P_n]$

Delay and hop count are examples of an additive parameter, while packet loss is an example of probabilistic, and delay variation is an example of root-mean-square basis. Bandwidth may be an example of a concave basis.

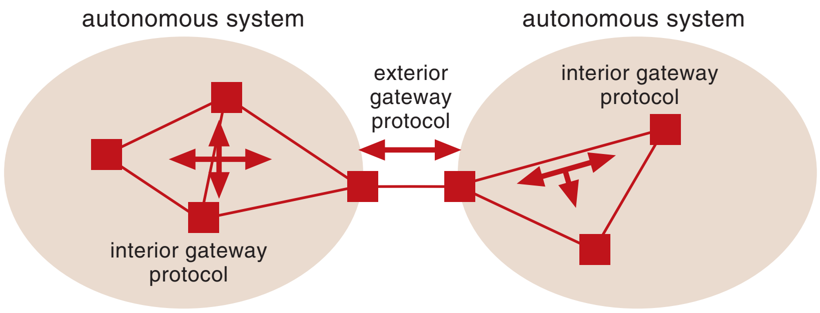

autonomous system
autonomous system
exterior gateway protocol
interior gateway protocol
interior gateway protocol

Figure 18 Use of exterior and interior gateway protocols

6.2.2 Routing Protocols

The original core routers in the ARPANET used a protocol call the Gateway-to-Gateway Protocol (GGP) to exchange routing information (every router was then referred to as a gateway). A router would exchange information with every neighbour router (which was fixed). The routing information consisted of a set of pairs (network, distance). The distance gives the cost of reaching that network. Here, cost was understood as the number of hops, meaning that low bandwidth paths with fewer hops would be preferred to higher bandwidth paths with more hops.

A set of routers can be grouped into an *autonomous system* (AS). An AS is handled by a single administrative authority, e.g. a network operator. A conceptual view on two ASs using Exterior Gateway Protocols (EGPs) between them and Interior Gateway Protocols (IGPs) internally is given in Figure 18. As recognised, a gateway/border router may use two different protocols, one within the AS and another outside the AS.

One of the first IGPs is the Routing Information Protocol (RIP), originally designed to provide consistent routing and reachability information among hosts in a local network at the University of California at Berkeley. Using RIP, routing data for each router is broadcast to all its neighbours periodically. Each destination in the routing table is included in the route updates. For a larger network, slow convergence may well occur, for instance when a network portion suddenly becomes unavailable (slow count to infinity). This could be alleviated by principles called split horizon and hold down, see [Come88]. Other IGPs are OSPF and IS-IS.

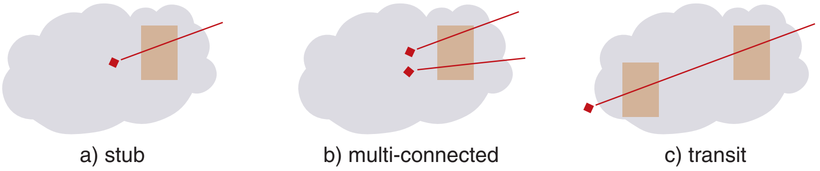
Two routers that belong to different ASs are said to be exterior neighbours. The protocol exterior neighbour used to advertise reachability information to other ASs is called the Exterior Gateway Protocol (EGP). An EGP has three main features: i) support a mechanism allowing two routers to agree to communicate reachability information (acquisition); ii) a router tests whether its EGP neighbour is responding; and iii) EGP neighbours exchange reachability information. The reachability information is commonly called routing information as it is used as a basis for deciding upon routing of packets. In order to fulfil its three features a number of message types are devised, like ‘acquisition’, ‘cease’, ‘hello’, ‘poll’ and ‘routing update’.

When two routers agree to exchange reachability information (acquisition), they also set initial values for a time interval to be used for testing whether the neighbour is alive (called a hello interval) and a polling interval that controls the maximum frequency of routing updates. These intervals can be changed. Moreover, they may be asymmetric, i.e. different values in the two directions. Considering features ii) and iii) above, one recognises that the reachability exchange has been separated from the routing information exchange. A motivation for this is that reachability could change more frequently without influencing the routing.

In a sense, EGP routing update messages can be considered as a generalisation of GGP routing updates as they include multiple routes (compared to a single route in the GGP). Basically, by using the routing information conveyed by EGP, a tree structure can be composed for each router where the router forms the root.

Between ASs, the Border Gateway Protocol (BGP) is commonly used. As seen from a BGP router, the network is made up of other interconnected BGP routers. Two routers are connected if they share a common network. Three categories of networks are used, see Figure 19: i) stub network that has only one connection to the BGP graph (no transit possible); ii) multiconnected network (could be used for transit, but does not allow it); and iii) transit network, which is used for transiting packets.

Figure 19 Network categories, referring to BGP



BGP can be said to be a distance vector protocol. In addition to the cost to each destination, each router does also keep information on the exact path to be used. Information on these exact paths is then exchanged. More information on BGP is found in [RFC1771] and [RFC1268].

BGP version 4 includes mechanisms that allow aggregation of routes and advertising of IP address prefixes. In one respect, one can say that

BGP applies a “hop-by-hop” routing; a BGP router only advertises the routes to its neighbours (in neighbouring domains) that it uses itself. BGP runs over TCP (see Section 3.4) and uses TCP port 179. After establishing a transport connection, BGP exchanges the initial data flow, which is the entire BGP routing table. Then, incremental updates are sent when something in the routing table is changing. Hence, periodical updates of the total routing table are avoided to save transfer capacity. Periodic “keep-alive” messages are however used to ensure that the peer BGP process is still running.

When transit is allowed, the routing information has to be conveyed between the border nodes. An example is the two nodes depicted in Figure 19 c). An interior “version” of BGP may then be used, referred to as IBGP, not treated in any nodes on the path between the pair of border nodes.

6.2.3 Routing and Traffic Engineering

From the routing perspective, networks are divided into Autonomous Systems (ASs) where each AS is divided into Interior Gateway Protocol (IGP) areas to allow for hiding and aggregating routing information. This way of hierarchical routing allows for more efficient routing handling, although from a traffic engineering perspective it may hide information, e.g. on paths used. Related to establishment of Label Switched Paths (LSPs) such information could be requested, leading to the introduction of additional features into the routing protocols, ref. [Jens01], e.g. to support traffic engineering.

Typical attributes identified to support traffic engineering operations are:

- maximum bandwidth;
- maximum reservable bandwidth;
- unreserved bandwidth (could be specified per class);
- resource class/colour.

These can be exchanged by the routing protocols in order to allow for constraint-based routing of LSPs. When LSPs are established by signalling, the protocols may be enhanced in order to take into account the constraints. In particular, when backup LSPs are to be set up, one should see to it that the backup path does not have overlapping hops with the primary path. This could be a challenging problem in particular when fibre optic cables carrying multiple wavelengths are used. Then the routing process should be informed of the grouping (i.e. the ones passing on the same cable). One suggestion is to use the resource class/colour field to indicate links that belong to the same group (goes on the same cable) [ID_ppro]. Such a grouping could also be utilised to reduce the amount of routing information to be exchanged, as similar routing measures may be applicable for all links in the group. This is also related to constraint-based routing as described in [Feng01].

7 Concluding Remarks

The main objective of this paper was to present formats and mechanisms related to the major protocols in Internet. These are the IP and the TCP, although UDP is gaining stronger foothold for traffic flows carrying timing sensitive data (audio and video). Several of the accompanying papers in this issue of the *Telektronikk* refer to protocol fields and procedures mentioned above.

References

[Come88] Comer, D. *Internetworking with TCP/IP.* Prentice-Hall, 1988.

[Feng01] Feng, B et al. State-of-the-art of IP Routing. *Telektronikk*, 97 (2/3), 130–144, 2001. (This issue.)

[ID_ecn] Ramakrishnan, K K et al. *The addition of Explicit Congestion Notification (ECN) to IP.* draft-ietf-tsvwg-ecn-00.txt, Nov. 2000. Work in progress.

[ID_pilc] Balakrishnan, H, Padmanabhan, V N. *TCP Performance Implications of Network Asymmetry.* draft-ietf-pilc--asym-02.txt, Nov. 2000. Work in progress.

[ID-ppro] Dovolsky, D, Bryskin, I. 2000. *Calculating protection paths and proxy interfaces in optical networks using OSPF.* draft-dovolsky-bryskin-cspf-pathprotect-proxy-00.txt. Work in progress.

[ID_tfrc] Handley, M, Padhye, J, Floyd, S. *TCP Friendly Rate Control (TFRC): Protocol Specification.* draft-ietf-tsvwg-tfrc-00.txt, Nov. 2000. Work in progress.

[Jens01] Jensen, T. Basic IP-related Mechanisms. *Telektronikk*, 97 (2/3), 54–85, 2001. (This issue.)

[Jens01a] Jensen, T. Network Principles and Applications. *Telektronikk*, 97 (2/3), 287–310, 2001. (This issue.)

[RFC793] IETF. 1981. Postel, J. *Transmission Control Protocol. DARPA Internet Program. Protocol Specification.* (RFC 793.)

[RFC1122] IETF. 1989. Braden, R. *Requirements for Internet Hosts – Communication Layers.* (RFC 1122.)

[RFC1268] IETF. 1991. Rekhter, Y, Gross, P. *Application of the Border Gateway Protocol in the Internet.* (RFC 1268.)

[RFC1323] IETF. 1992. Jacobson, V, Braden, R, Borman, D. *TCP Extensions for High Performance.* (RFC 1323.)

[RFC1644] IETF. 1994. Braden, R. *T/TCP. TCP Extensions for Transactions Functional Specification.* (RFC 1644.)

[RFC1771] IETF. 1995. Rekhter, Y, Li, T. *A Border Gateway Protocol 4 (BGP-4).* (RFC 1771.)

[RFC1981] IETF. 1996. McCann, J, Mogul, J, Deering, S. *Path MTU Discovery for IP version 6.* (RFC1981.)

[RFC2001] IETF. 1997. Stevens, W. *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms.* (RFC 2001.)

[RFC2402] IETF. 1998. Kent, S, Atkinson, R. *IP Authentication Header.* (RFC 2402.)

[RFC2406] IETF. 1998. Kent, S, Atkinson, R. *IP Encapsulating Security Protocol (ESP).* (RFC 2406.)

[RFC2460] IETF. 1998. Deering, S, Hinden, R. *Internet Protocol, Version 6 (IPv6) Specification.* (RFC 2460.)

[RFC2463] IETF. 1998. Conta, A, Deering, S. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.* (RFC 2463.)

[RFC2988] IETF. 2000. Bernet, Y et al. *A Framework for Integrated Services Operation over Diffserv Networks.* (RFC 2998.)

[Tane96] Tanenbaum, A S. 1996. *Computer Networks.* Upper Saddle River, NJ, Prentice Hall.

Traffic Engineering Principles, Activities and Mechanisms

TERJE JENSEN

Moving beyond the single class, best-effort IP network, most operators introduce Traffic Engineering (TE) mechanisms. These mechanisms are fairly crucial for further operation, supporting the portfolio of services requested.

This article gives an overview of TE concepts and mechanisms by describing taxonomy and organisation of TE activities. It is mainly drawing on results presented in an Internet draft (ref. [ID_tepri]).

Terje Jensen (39) is Research Manager at Telenor R&D, Kjeller. He earned his PhD degree in 1995 from the Norwegian University of Science and Technology. Activities include performance modelling and analysis, dimensioning and network evolution studies.

terje.jensen1@telenor.com

1 Introduction

A basic definition of Traffic Engineering (TE) is

> *performance optimisation of operational networks, including measurement, modelling, characterisation and control.*

This basically means that running networks are in focus; however, also longer-term planning has to be considered for a running network to be able to cope with the future traffic flows and their characteristics.

Key performance objectives of Traffic Engineering (TE) can be categorised, ref. Figure 1, as

- traffic oriented; or
- resource oriented.

The first includes means undertaken to improve the provision of services, having objectives like reduced delay, reduced packet loss, increased throughput. Resource oriented objectives optimise the resource utilisation, resulting in less installed network capacity. Trade-offs are commonly seen between these two perspectives.

The traffic flows to be served by the network will likely have a range of requirements. Two types of requirements are delay and loss. To a certain extent, these may also face a trade-off as shown in Figure 2.

Hence, for a given traffic load there may be an option to "trade" between delay and loss by adapting the buffer size, as long buffers give higher delays but lower losses. By reducing the traffic load, both loss and delay decrease. Typically, real-time traffic flows ask for fairly low losses and low delays (and low delay variations). Hence, shorter buffers and lower traffic loads can be used as thresholds for such flows. On the other hand, when more elastic traffic flows are

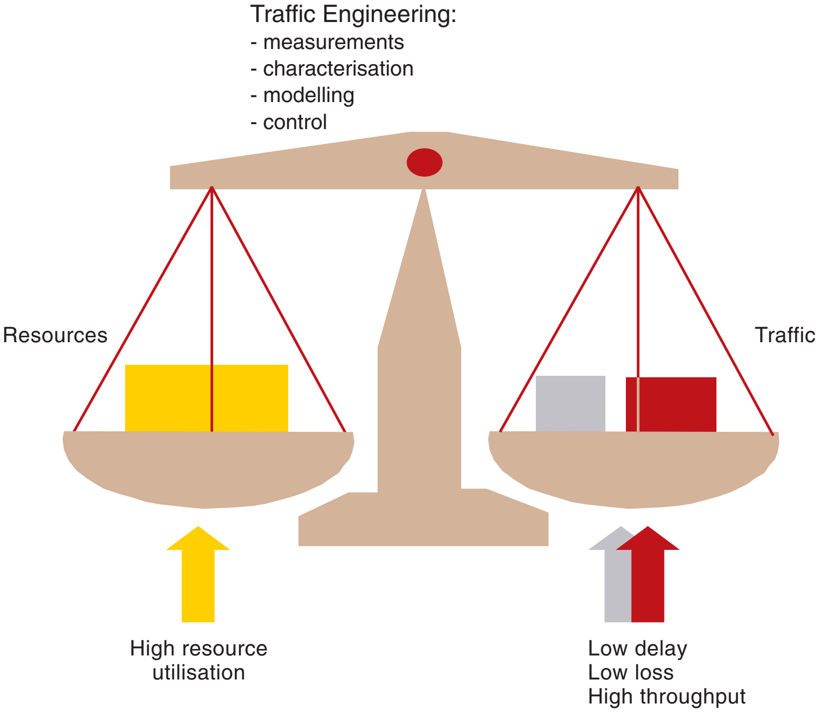


Figure 1 Balancing traffic and resource concerns

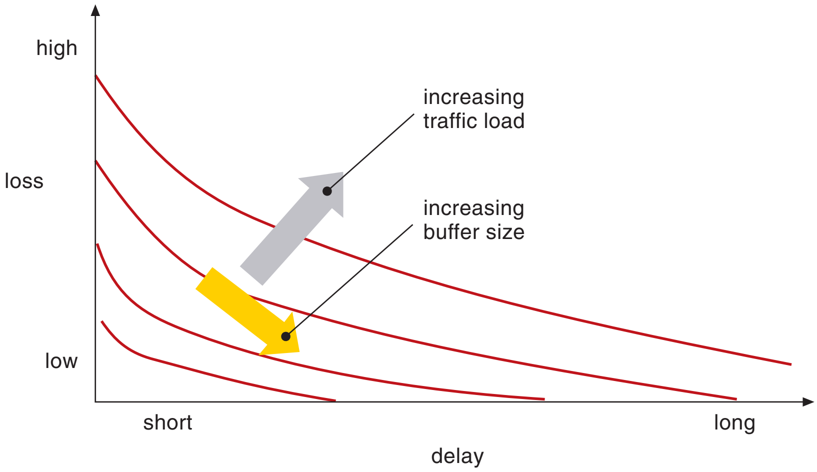


Figure 2 Schematic illustration of relations between delay and loss

served, longer buffer sizes may apply as these would accept longer delays (and delay variations), although the loss requirements might not be lower.

One of the primary goals of TE in an operational network is to limit the sustained congestion. A focused congestion may result from unbalanced mapping of traffic flows onto the resource groups. Then, means can be activated to distribute the traffic flow in a better way. TE mechanisms do also allow for differentiating and ensuring service levels. This would meet the characteristics related to the spectrum of traffic flows. An example is to use separate real-time and non-real-time traffic on the buffering side, while still using the same link capacity. Hence, high link utilisation is achieved, while knowledge of the traffic flow characteristics is exploited. Delay, delay variation and loss ratio are examples of characteristics used so far. Another important parameter is related to dependability, such as the availability of the service. This would also be addressed by the TE-related mechanisms, increasing the general dependability but also allowing for differentiation.

In sum, the TE mechanisms offer a set of “tools” that a network operator can tune for its operation, reaching better utilisation of network resources while allowing predictable service levels (differentiated and ensured).

The actual need for introducing TE-related mechanisms is questioned by some. Two factors supporting this view are the traffic growth and the willingness to pay. For the former, it is considered that by the traffic growth seen these days there is little need for accurate procedures as more capacity should be installed all the time. Hence, additional capacity present when over-provisioning would fairly soon be needed anyway. Regarding the latter factor, much traffic on IP-based networks comes from Web browsing, assumed to be looking for low-priced, low service level.

An argument in the other direction is that as one of the current trends is that more commercial activities are based on IP networks, a stronger demand for predictable services will emerge. This also seems to be recognised by the industry in this area as much work is allocated to TE-related mechanisms.

In this article, some of the most central themes of TE are described. In the following chapter overall objectives, scope and resource types are described. Chapter 3 presents the activities, processes, key components and mechanisms together with the contexts in which the TE activities are carried out. Some requirements on TE systems are listed in Chapter 4. Chapter 5 explains the TE taxonomy, while Chapter 6 elaborates on further challenges in the TE and QoS areas.

2 TE Objectives and Resource Types

The main goals of traffic engineering are to improve performance for IP-based traffic while still utilising the network resources efficiently. In the TE framework principles, as elaborated within IETF, architectures and methodologies for evaluating and optimising the performance of operational IP networks are addressed. The framework as described in [ID_tepri] gives both the terminology (set of key terms) and the taxonomy (criteria for describing a system). Internet traffic engineering is here defined as

> *that aspect of Internet network engineering dealing with the issue of performance evaluations and performance optimisation of operational IP networks. Hence, measurement, characterisation, modelling and control of traffic are included.*

A major objective is to improve the performance of operational networks, at the traffic and at the resource levels. This is striven for by looking at traffic-related performance requirements, like delay, delay variation, packet loss and goodput. At the same time the network resources should be efficiently utilised. An essential part is to achieve reliable network operations, e.g. during failures. Having efficient routing configurations is also central as these decide the way the packets are distributed in the network.

Capacity management and traffic management can be used for the optimisation done as part of TE. Capacity arrangement includes capacity planning, routing control and resource management. The resources include *link bandwidth, buffer space* and *computational*, see Figure 3.

Traffic management includes nodal traffic control (e.g. traffic conditioning, queue management, scheduling) and other functions that regulate traffic through the network or control access to network resources. These activities can be carried out continuously and in an iterative manner. The activities are commonly divided into proactive and reactive. In the former preventive and perfective actions can be found, while corrective actions would be part of the latter.

The TE actions would operate on various time scales; coarse (days, years – like for capacity management), intermediate (ms, days – like for routing control), and, fine (ps, ms – like for packet level processing).

The TE subsystems include capacity augmentation, routing control, traffic control and resource control. Inputs to the TE system would include network state variables, policy variables and decision variables. A challenge is to introduce automated capabilities that adapt fast and efficiently to changes in the network state, while stability is maintained. Performance evaluation is then a critical part of this, to assess the effectiveness of a TE method, to monitor a network state and to verify compliance with performance levels.

3 TE Settings and Activities

3.1 Settings

A number of settings for exercising TE activities are identified in [ID_tepri]. To some extent these can also be considered as steps, see Figure 4. However, considering that TE activities are carried out continuously the different steps may be active at the same time, although possibly looking at different instances of time for implementing the solutions into the network.

The settings described are:

- *Network* context; describing the situations where traffic engineering challenges are found. Such situations include network structure, network policy, network characteristics, network constraints, network quality attributes, network optimisation criteria, etc.
  A network can be represented as a system, see Figure 5, consisting of: i) a set of interconnected resources; ii) a demand representing the offered load; and iii) a response consisting of network processes, protocols and mechanisms that carry the offered load through the network. All these elements may have specific characteristics, which for example may limit the flexibility. Several types of demand classes may be present, similar to traffic classes although also different customer types should be taken into account. This results in a request for differentiated services. The network resource and the traffic handling-related mechanisms also have their characteristics. Some detail of how the network provides the services will be given by the policies specified. In [ID_tepri] it is stated that requirements on the service provision (traffic handling) can either be statistical (e.g. by rates and burst sizes) or deterministic (e.g. some effective bit rate measure). Requirements on the QoS are either of the integrity type (e.g. packet loss) or of temporal nature (e.g. delay, delay variation).

- *Problem* context; defining the issues that TE addresses, like identification, abstraction, representation, formulation, requirement specification, solution space specification, etc. One class of problems is how to formulate the questions that traffic engineering should solve; how to describe requirements on the solution space, how to describe desirable features of good solutions, how to solve the problems and how to characterise and measure the effectiveness of the solutions. Another problem is how to measure and assess the network state parameters, including the network topology. A third class of problems is how to characterise and evaluate network states under a variety of scenarios. This can be addressed both on system level (macro states – “macro TE”) and resource level (micro state – “micro TE”). This asks for appropriate levels of abstractions being identified. A class of prob-

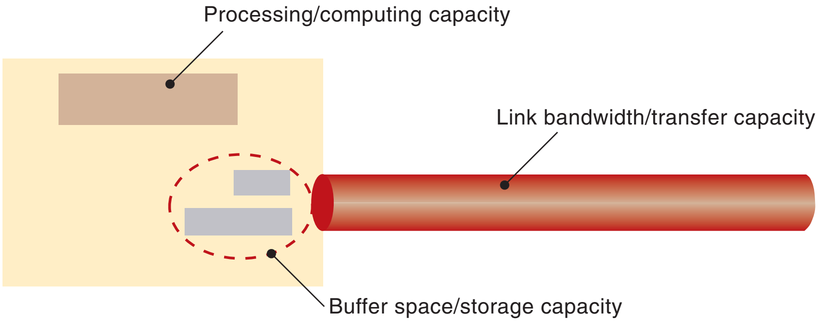

Processing/computing capacity
Link bandwidth/transfer capacity
Buffer space/storage capacity

Figure 3 Basic resource types related considered for Traffic Engineering

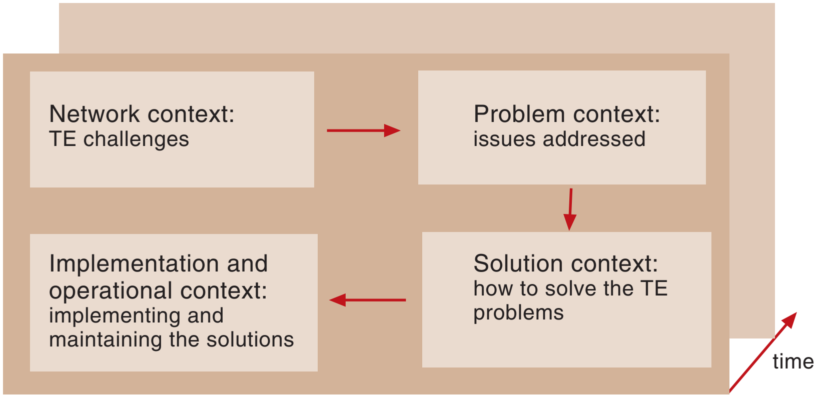

Network context:
TE challenges
Problem context:
issues addressed
Solution context:
how to solve the TE problems
Implementation and operational context:
implementing and maintaining the solutions
time

Figure 4 Settings for exercising TE activities

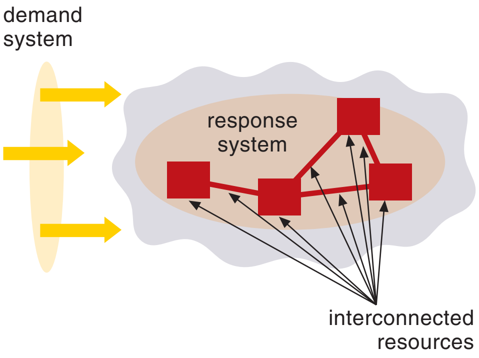


Figure 5 Elements in the network context

lems is also how to optimise the performance of a network. Solving congestion is an essential part of performance improvement. Handling congestion can be divided into demand side policies (restrictive) and supply side policies (expansive).

- *Solution* context; elaborating how to solve the TE problems. This includes evaluation of alternatives. This requires estimating traffic load, characterising network state, elaborating solutions on TE problems and setting up a set of control actions. The instruments relevant include: i) set of policies, objectives and requirements for network performance evaluation and optimisation; ii) set of tools and mechanisms for measurement, characterisation, modelling and controlling traffic and allocation to network resources; iii) set of constraints on the operating environment, network protocols and TE system; iv) set of quantitative and qualitative techniques and methods for abstracting, formulating and solving TE problems; v) set of administrative control parameters that may be managed by a configuration management system; vi) set of guidelines for network performance evaluation, optimisation/improvement. Traffic estimates can be derived from customer subscription information, traffic projections, traffic models, and from empirical measurements. Polices for handling the congestion problem can be categorised according to the criteria: i) response time scale (long – weeks to months, e.g. capacity planning; medium – minutes to days, e.g. setting routing parameters, adjusting Label Switched Path (LSP) design; short – ps to minutes, e.g. packet processing of marking, queue management); ii) reactive versus preventive; and iii) supply side (increase available capacity, redistribute traffic flows) versus demand side (control the offered traffic).

- *Implementation and operational* context; implementing the actual solutions, involving planning (including a priori to determine actions based on triggers), organisation (including assigning responsibilities to different units and co-ordinating activities), and execution (including measurement and application of corrective and perfective actions).

These context descriptions may also be looked upon as gradually getting more precise and closer to the implementation.

3.2 TE Process Model

A TE process model is presented in [ID_tepri]. This is depicted in Figure 6 as an iterative procedure consisting of four main steps.

The first phase includes definition of *control policies*. These would typically depend on a set of inputs, like business model, network cost structure, operating constraints, utility model and optimisation criterion.

The second phase involves *measurements* in order to assess the conditions in the network; network state and traffic load.

The third phase consists of *analysing the network state and characterising the traffic load.* A number of potential models and analysing techniques may be relevant, for instance also looking at the timely and spatial distribution of the traffic load.

In the fourth phase, *performance optimisation* is done. This includes a decision process selecting and implementing a set of actions. Actions may work on the load demand, distribution of load and network resource configuration and capac-

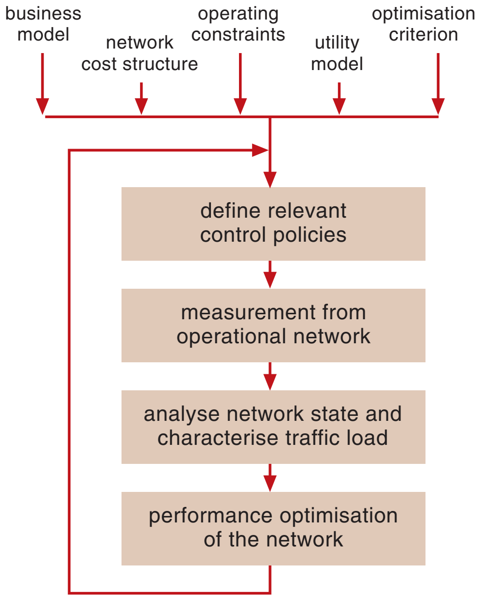


Figure 6 TE process model

ity. This may also initiate a network planning in order to improve network design, capacity, technology, and element configuration.

The actual relations between the process model and the context are not elaborated on in [ID_tepri]. However, considering that an early phase is to assess conditions of the network, most of the process model relates to an operational network. The context/setting would also include perspectives on a longer time, e.g. allowing for more aspects to be considered.

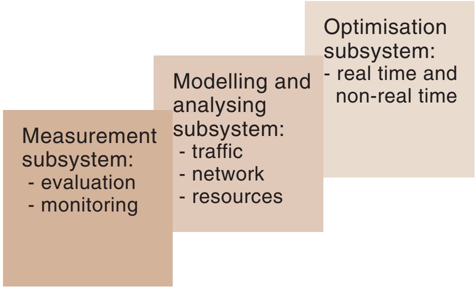


Figure 7 TE key components/ subsystems

3.3 TE Key Components

The key components of the TE process model are (see Figure 7):

- *Measurement* subsystem: Carrying out measurement is essential to providing feedback on the system state and performance. It is also critical in order to assess the service level provided (and QoS) and effect of TE actions. A basic distinction between monitoring and evaluation is to be observed; monitoring refers to the provision of raw data, while evaluation refers to the use of the raw data for inferring on the system state and performance. Measurements can be carried out at different levels of aggregation, e.g. packet level, flow level, user level, traffic aggregate level, component level, network-wide level, and so forth. In order to perform measurements systematically, several questions have to be answered, like [ID_tepri]: Which parameters are to be measured? How should the measurements be accomplished? Where should the measurements be performed? When should the measurements be performed? How frequently should the monitored variables be measured? What level of measurement accuracy and reliability is desirable and realistic? To what extent can the measurement system permissibly interfere with the operational network conditions? What is the acceptable cost of measurements?

- *Modelling and analysis* subsystem: A central part of the modelling is to elaborate a representation of the relevant traffic characteristics and network behaviour. In case a structural model is used, the organisation of the network and its components are the main emphasis. When behavioural models are used, the dynamics of the network and traffic are the key issues. The latter model is particularly relevant when performance studies are undertaken. Then adequate models of the traffic sources are also needed.

- *Optimisation* subsystem: Optimisation can be categorised as real-time and non-real-time. The former operates on short to medium time scales (e.g. ms to hours) and works on adjusting parameters in mechanisms in order to relieve congestion and improve performance. Examples of means are tuning of routing parameters, tuning of buffer management mechanisms and changing Label Switched Paths (LSPs). Non-real-time is also seen as network planning, typically working on a longer scale. For both of these, stability and robustness are essential concerns.

Routing is a central component in efficient handling of traffic flows in an IP-based network. When introducing a number of service classes, some additional constraints can also be considered when deciding upon the possible routing. Examples of such constraints are available bandwidth, hop count, and delay. This implies that possible paths as seen from a router must have the corresponding attributes attached.

3.4 Mechanisms and Subjects

In order to complement the best effort service, a number of activities are undertaken by different IETF groups as well as by others. The subjects listed below are treated more in detail in accompanying papers of this *Telektronikk* issue:

- Integrated Services *(IntServ)*. Applying this service model requires that resources are reserved before the traffic flow starts. As mentioned earlier, transmission links and buffers are commonly seen as resources. Mechanisms like packet classifiers, packet schedulers and admission control units have to be present in the routers supporting IntServ. A classifier identifies flows that are to be served with a certain level. A scheduler handles the service scheduling to ensure that requirements of the traffic flow are met. Admission control determines whether or not a router has the needed resources available to accept a new flow while still meeting all requirements for all flows present. Two additional service classes are identified: guaranteed service and controlled-load service. As state information has to be kept for each group of traffic flows, a router in a larger network may have capacity problems keeping all that information. Hence, IntServ is frequently claimed to face the scalability prob-

Box A Scalability

The term scalability is frequently observed in the discussion on various mechanisms. Looking up in a dictionary, scalabiltiy is simply described as to allow increase (or decrease) of something.

A common interpretation of scalability, however, is that the network equipment may not be able to cope with the growing number of traffic flows or other units used to express the amount. Hence, there is a threshold on the number of units that a network node can handle. Such an understanding may implicitly assume that the operating point is known (i.e. the range of number of units is given). Furthermore, modifying the network node, e.g. by upgrading, would also affect the capacity threshold.

A statement seen is that IntServ does not scale well, although this will likely follow a linear growth as a function of the number of flows, see Figure Box A-1. DiffServ, on the other hand, has an improved scalability as aggregates are looked at. Introducing MPLS in combination with DiffServ may result in higher demands on the nodes.

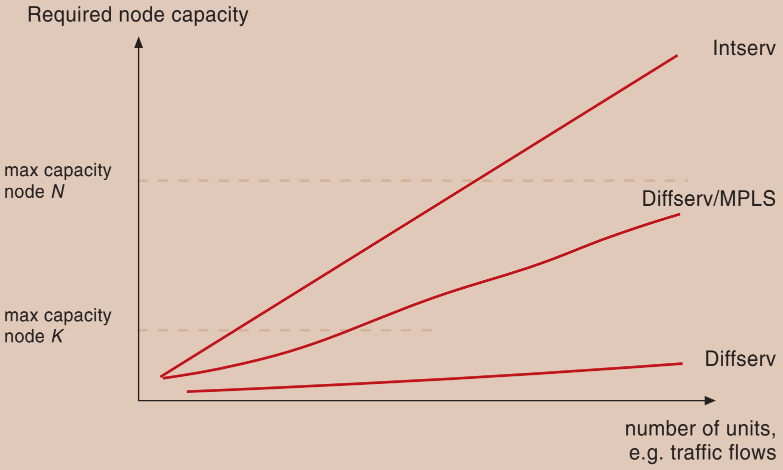


Figure Box A-1 Scalability concerns – for illustration only

Interpreting scalability may therefore ask for a certain tolerance in the sense that inherent demands on the network nodes capacity are assumed.

lem (see Box A). Furthermore, the reservation information has to be conveyed between the routers. The Resource reSerVation Protocol is one means of doing this.

- Resource reSerVation Protocol *(RSVP)*. RSVP is a signalling protocol allowing a receiver to initiate establishment of the reservation. It is a so-called soft state protocol in the sense that the reservation has to be refreshed repeatedly in order to keep the reservation for a longer interval. Both multicast and unicast flows are supported.

- Multi-Protocol Label Switching *(MPLS)*. MPLS can be said to provide an additional forwarding mechanism. At the ingress of an MPLS domain, Label Switching Routers (LSRs) classify IP packets into Forwarding Equivalence Classes (FECs) based on certain criteria. An MPLS label is then prepended to the packets. Then, the subsequent LSRs may only look at the MPLS label to decide upon the forwarding treatment. A Label Switched Path (LSP) is the path between an ingress LSR and an egress LSR on which the packets are sent. LSPs can be used for several purposes, including load distribution, Virtual Private Networks, and multicasting.

- Differentiated Services *(DiffServ)*. Addressing the scalability challenge related to IntServ, DiffServ was proposed to categorise the traffic flows into a limited set of service classes. A class is defined using different classification, policing, shaping and scheduling policies. Hence aggregates of traffic flows are used, alleviating intermediate routers to consider individual traffic flows. A DiffServ field is defined in the IP header (part of the ToS octet, see [Jens01]) in order to indicate which service class a packet belongs to.

- *Measurements*. A set of metrics is developed by the IETF IP Performance Metrics (IPPM) working group. These can be used to monitor the performance observed by the end-users, network operators, and others. An architecture for handling measurements has been made by the IETF Real Time Flow Measurement (RTFM) working group. This architecture defines methods to do measurements of traffic flows and components involved. Such a system consists of meters, meter readers and managers.

- End-point *congestion management*. The IETF End-point Congestion Management working group is set to define congestion control mechanisms that transport protocols can use. A congestion manager monitors the paths of every congestion group under its control, using this information to distribute the capacity among the traffic flows in the group. Besides, procedures defined as part of TCP do also address congestion control, ref. [Jens01].

4 Requirements on TE Systems

[ID_tepri] describes a number of requirements that a TE system should fulfil. Here a requirement is understood as a capability needed to solve a TE problem or to achieve a TE objective. The requirements are either non-functional or functional. A non-functional requirement relates to the quality attributes of state characteristics of a TE system. A functional requirement gives the function a TE system should perform in order to reach an objective or address a problem.

4.1 Non-functional Requirements

The generic non-functional requirements given in [ID_tepri] are:

- Usability. This is a human factor aspect referring to the ease of deployment and operation of a TE system.
- Automation. Commonly, as many functions as possible should be automated, reducing the human effort to control and analyse the information and network state. This is even stronger for a larger network.
- Scalability. The TE system should scale well when the number of routers, links, traffic flows, subscribers, etc. grows. This may imply that a scalable TE architecture is applied.
- Stability. This is an essential requirement for an operational system avoiding adverse results for certain combinations of input and state information.
- Flexibility. A TE system should be flexible both in terms of the optimisation policy and the scope. An example of scope is that additional classes should be considered in case these are introduced into the network. Another aspect of flexibility is that some subsystems of the TE system could be enabled/disabled.
- Visibility. Mechanisms to collect information from the network elements and analyse the data have to be present in a TE system. These would then allow for presenting the operational conditions of the network.
- Simplicity. A TE system should be as simple as possible, that is considered from the outside, not necessarily using simple algorithms. Simplicity is particularly important for the human interface.
- Efficiency. As little demanding, in terms of processing and memory resources, as possible is requested. However, this also refers to the result from the TE system being obtained in a timely manner.
- Reliability. A TE system should be available, in the operational state, when needed.
- Survivability. Recovering from a failure and maintaining the operation is requested, in particular for the more critical functions of a TE system. Commonly, this requires that some redundancy is introduced.
- Correctness. A correct response (according to the algorithms implemented) has to be obtained from a TE system.
- Maintainability. It should be simple to maintain a TE system.
- Extensibility. It should be easy to extend a TE system, e.g. when introducing new functions and when the underlying network is extended.
- Interoperability. Open standards should be used for the interfaces in order to simplify interoperation with other systems.
- Security. Means supporting integrity, information concealment, etc. have to be implemented.

As mentioned, some of these requirements may be mandatory while others are optional for a particular TE system.

4.2 Functional Requirements

Some functional requirements are also described in [ID_tepri], such as those related to:

- Routing. An efficient routing system should take both traffic characteristics and network constraints into account when deriving the better routing schemes. A load splitting ratio among alternative paths should be configurable allowing for more flexibility in the traffic distribution. Some routes of subsets of traffic should also be controllable without affecting routes of other traffic flows. This has particular relevance when several classes are present in the network. In order to convey information on topology, link characteristics and traffic load, several of the relevant routing protocols have to be enhanced. An example is constraint-based routing which is gaining

Box B Protection Types

Protection types for MPLS networks are categorised as: i) link protection (backup LSP is disjoint from the working LSP at the particular link for which protection is required), being a link local repair; ii) node protection (backup LSP is disjoint from the working LSP at the node considered); iii) path protection (protect a working LSP from failure at any point along its path, hence the back up and working LSPs are both node and link disjoint); and iv) segment protection (failure within a domain is corrected within that domain).

Segment protection can also be to reroute an LSP locally, e.g. between the routers connected to the failed link (as opposed to end-to-end rerouting of an LSP).

Protection options are typically given by *m:n*, where *m* refers to the number of working LSPs to be protected and *n* refers to the number of backup LSPs. Common combinations seen are 1:1, *k*:1, and 1:*k*. The second can be used when the traffic load of the working LSP is to be distributed, e.g. due to bandwidth requirements. A protection option called 1+1 is also used when the traffic is sent both on the working LSP and the backup LSP. Then the egress LSR selects one of the two LSPs.

more interest. This addresses the selection of paths for packets and may work well with path-oriented solutions, that is LSPs.

- Traffic mapping. This refers to the assignment of traffic flows onto paths to meet certain requirements considering the set of policies relevant. A central issue arises when several paths are present between the same pair of originating and destination router. Appropriate means should be taken to distribute the traffic according to some defined ratios, still keeping the ordering of packets belonging to the same application (or micro-flow).

- Measurement. Mechanisms for monitoring, collecting and analysing statistical data have to be in place. Such data may relate to performance and traffic. In particular, being able to construct traffic matrices per service class is an essential part of a TE system.

- Network survivability. Survivability refers to the capability to maintain service continuity in presence of faults. This can be realised by rapid recovering or by redundancy. Co-ordinating protection and restoration capabilities across multiple layers is a challenging task. At the different layers protection and restoration would typically occur at different temporal and bandwidth granularity. At the bottom layer, an optical network layer may be present, e.g. utilising WDM. Then, SDH and/or ATM could be present below the IP layer. Restoration at the IP layer is commonly done by the routing protocols, which may require some minutes to complete. Some means being proposed relate to MPLS allowing for faster recovery (ref. Box B). A common suit of control plane protocols has been proposed for the MPLS and optical transport networks. This may support more sophisticated restoration capabilities. When multiple service classes are present, their requirements on restoration may differ introducing further challenges on the mechanisms to be used. Resilience attributes can be attached to an LSP telling how traffic on that LSP can be restored in case of failure. A basic attribute may indicate if all traffic trunks in the LSP are transferred on a backup LSP or some of the traffic is to be routed outside, e.g. following the routing protocols (see Box C). Extended attributes may be introduced giving indications like backup LSP is to be pre-established, constraints for routing the backup LSP, priorities when routing backup LSP, and so forth.

- Servers and content distribution. Location and allocation of content on servers have significant impact on the traffic distribution, in particular as long as much of the traffic is similar

Box C Resilience

As the multitude of services on IP-based networks increases, a differentiation of dependability (e.g. reliability and availability) requirements is asked for. This is sometimes referred to as resilience requirements. A resilience differentiated network would only protect traffic flows that require higher availability that would allow for more effective network resource utilisation. Basically, this could be done on other layers than IP. In principle realising resilience mechanisms in lower layers would allow for faster response, e.g. from a link is broken until an alternative link is found. On the other hand, lower layers also operate on more coarse granularity of traffic aggregates. Thus, higher layers give finer granularity but commonly also longer response times.

Traffic flows requiring high availability may belong to so-called mission-critical applications. Such applications may include all types of applications, like real-time, elastic, etc. Therefore, applications such as multimedia as well as database transactions could be asking for high availability. This means that the resilience requirement is orthogonal to other performance variables, like delay and loss.

A possible set of resilience classes is described in [ID_resreq]:

- High resilience requirements: Resources should be exclusively reserved on an alternative path. For a 1+1 protection, packets are forwarded on the working path and the backup path. In case the working path fails, the receiver just selects packet on the other path. In case of 1:1 protection, the packets are forwarded on the alternative path in case of failure on the working path. This asks for recovery signalling to handle unidirectional failures.
- Medium resilience requirements: Spare resources may be shared between multiple flows. The bandwidth management has to assure that enough spare resources are available for a given set of expected failures. In case of a failure, packets are forwarded after rerouting and reservation of spare resources.
- Low resilience requirements: No resources are reserved in advance. In case of failure, packets may be forwarded after a rerouting and reservation phase if enough resources are available.
- No resilience requirements: In case of a network failure in the administrated domain, packets may be discarded/dropped. This may happen even if the traffic is not directly affected by the failure but rather by a rerouting of other traffic flows having high resilience requirements.

In order to implement differentiated resilience, updates of the service implementation could be needed. For IntServ/RSVP corresponding attributes have to be carried in the signalling messages and filters/conditioners have to be available. For DiffServ activating a resilience marking may be used, like a bit in the ToS octet (bit 5 of the DSCP field). Resilience attributes may also be used for MPLS. These mechanisms are described for the different mechanisms in this article.

Box D Information Distribution

In current IP-based networks several means are used to make the distribution of information more efficient, including:

- *mirroring* the information meaning that the information is replicated in several places/servers. This would increase the dependability and allow for faster response;
- *caching* the information, basically meaning that previously accessed information is stored in a place closer to the user (limiting the traffic and allowing faster response);
- *load balancing* having the objective to distribute the traffic/requests among the servers.

to client – server interactions. Hence, load balancing directing traffic on the different servers may improve the overall performance. This is sometimes called traffic directing, operating on the application layer, ref. Box D.

- DiffServ issues. As DiffServ is more widely deployed, adequate TE systems become more critical to ensure that conditions in Service Level Agreements (SLAs) are met. Service classes (Class of Service) can be offered by defining Per-Hop Behaviours (PHBs) along the path, exercising DiffServ in the nodes, in particular by configuring mechanisms like traffic classification, marking, policing and shaping (mainly in edge routers). A PHB is a forwarding treatment including buffer management and scheduling. In addition the amount of service capacity, e.g. bandwidth, allocated to the different service classes has to be decided upon. The following issues, from [ID_tepri], give some requirements on TE in a DiffServ/MPLS environment:

  - An LSP should provide configurable maximum reservable bandwidth and/or buffer for each supported service class.

  - An LSR may provide configurable minimum available bandwidth and/or buffer for each class on each of its links.

  - In order to perform constraint-based routing on a per-class basis for LSPs, the routing protocols should support extensions to propagate per-class resource information. When delay bounds is an issue, path selection algorithms for traffic trunks with bounded delay requirement should take delay constraints into account.

  - When an LSR dynamically adjusts resource allocation based on per-class LSP resource requests, adjusting weights for the scheduling algorithms should not adversely impact delay and jitter characteristics.

  - An LSR should provide configurable maximum allocation multiplier on a per-class basis.

  - Measurement-based admission control may be used to improve resource usage, especially for classes not having strict loss or delay/jitter requirements.

- Controlling the network. In order to see the effect of having a TE system, the relevant decisions must be introduced into the network. Control mechanisms may be manual or automatic, the latter being a goal for most. Network control functions must be secure, reliable and stable, in particular during failure situations.

5 TE Taxonomy

A taxonomy of TE systems is given in [ID_tepri] in accordance with the following criteria:

- *Time-dependent* vs. *state-dependent* vs. *event-dependent*: A static TE system implies that no TE methods are applied on the time scale considered. Therefore, it is commonly assumed that all TE schemes are dynamic (on the time scale looked at). A time-dependent scheme is based on timely variations in traffic patterns and used to pre-program changes in the traffic handling. A state-dependent scheme adapts the traffic handling based on state of the network, allowing for taking actual variations in the traffic patterns into account. The state of a network may be based on resource utilisation, delay measures, etc. Accurate information available is crucial for adaptive TE schemes. This information has to be gathered and distributed to the relevant routers. A challenge is to limit the amount of information that must be exchanged between routers, still allowing for sufficiently updated data in each of the routers to make the traffic handling decisions. Event-dependent schemes may lead to fewer information exchanges compared to state-dependent schemes. Then, certain events are used as input when updating the traffic handling, like traffic load crossing a threshold, unsuccessful establishment of an LSP, etc.

- *Offline* vs. *online*. In case changes in traffic handling do not need to be done in real time, the computations can be done offline, e.g. allowing for more thorough searches over the feasible solutions finding the better one to apply. On the other hand, when traffic handling is to adapt to changing traffic patterns, it is to be done online. For online calculations, relatively simple algorithms are applied leading to short response times until the updated traffic handling can be activated. Still the algorithm should present a solution that is close to the optimal one.

- *Centralised* vs. *distributed*. In a centralised scheme a central function decides upon the traffic handling in each of the routers. Then, the central function has to collect and return the information. In order to limit the overhead, infrequent information exchanges are sought; however, more frequent exchanges are asked for to keep an accurate picture of the network state in the central function. This results in a classic trade-off problem, finding the time interval for collecting and returning the information. A similar trade-off is also seen for the distributed scheme, although then the decisions are made by each router. A drawback of a centralised scheme is commonly that a single point of failure is introduced, implying that the central function is available and has sufficiently processing capacity for the scheme to work efficiently.

- *Local* vs. *global information*. Local information refers to a portion of the region/domain considered by the TE system. An example is delay for a particular LSP. Global information refers to the whole region/domain considered.

- *Prescriptive* vs. *descriptive*. When a prescriptive approach is used, a set of actions would be suggested by the TE system. Such an approach can be either corrective (an action to solve an existing or predicted anomaly) or perfective (an action suggested without identifying any particular anomaly). A descriptive approach characterises the network state and assesses the impact from exercising various policies without suggesting any specific action.

- *Open loop* vs. *closed loop*. In an open loop approach, the control actions do not use feedback information from the network. Such feedback information is used when a closed loop approach is followed.

- *Tactical* vs. *strategic*. A tactical approach considers a specific problem, without taking into account the overall solutions, tending to be ad hoc in nature. A strategic approach considers the TE problem from a more organised and systematic perspective, including immediate and longer-term consequences.

- *Intradomain* vs. *interdomain*. Interdomain traffic engineering is primarily concerned with performance of traffic and networks when the traffic flows cross a domain, e.g. between two operators. Both technical and administrative/business concerns make such a TE activity more complicated. One example is based on the fact that Border Gateway Protocol version 4 (BGP-4), being the (default) standard routing protocol, does not carry full information like an interior gateway protocol (e.g. no topology and link state information). In a business sense it would not be likely that two parties, being potential competitors, would reveal all that data of their network. Another aspect is the presence of relevant SLAs that govern the interconnection, including description of traffic patterns, QoS, measurements and reactions. An SLA may explicitly or implicitly specify a Traffic Conditioning Agreement (TCA, which defines classifier rules as well as metering, marking, discarding and shaping rules.

A specific TE system can then be categorised by applying the criteria listed above.

6 Further Issues

6.1 Basic Questions and Factors

Several forces are influencing on the evolution of IP-based networks, see Figure 8.

A number of basic questions can be raised related to the future of Internet/IP-based network:

- Will the current Internet routing mechanisms operate as steadily more hosts and networks are added?

- How is it possible to automate mechanisms for storing and locating information about individual users?

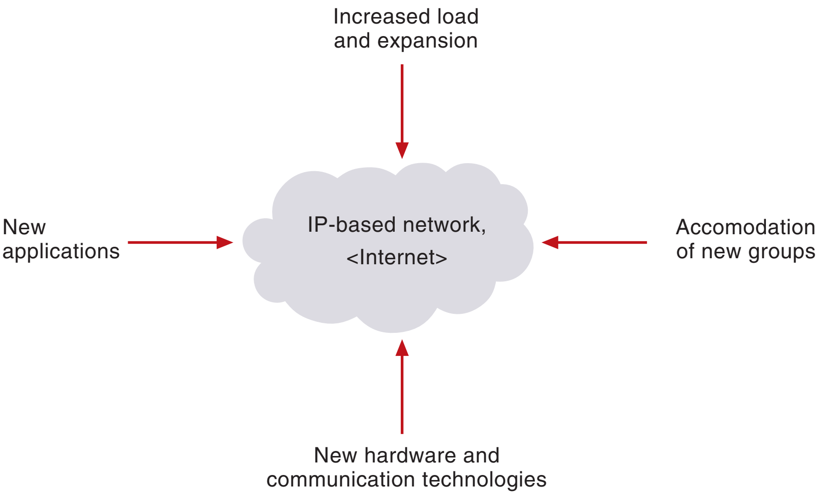
- How is it possible to automate mechanisms for storing and locating information about services offered by hosts?

- How to incorporate vendor-independent and automated mechanisms that allow monitoring and control of traffic and network resources?

- How can relevant protocols be adapted, or supplemented, to accommodate new applications that have specific requirements (e.g. high throughput, short delays)?

- How can emerging business configurations be supported, allowing for multiple services, access forms and a range of providers/operators.

Figure 8 Factors influencing the evolution of IP-based networks, adapted from [Come88]



6.2 Open Issues Related to QoS Architecture

As the best effort service is the most commonly seen in most IP-based networks, a wider range of service levels is also sought. The service levels can be widened in two respects; to provide a service level improved from best effort, and to provide a service level which is more predictable.

Basically, a few features have to be in place in order to allow for differentiating service levels. Firstly, the application has to convey the information to the network of which service level it wants (naturally, this may also be done manually). Then, resources in the network have to be available to provide the service level agreed. In order to ensure that resources are available, some load control and resource usage/configuration schemes have to be applied. If an application request the service level for a certain traffic flow, it also has to mark the flow in a way for the network to recognise it. Commonly, there are also some constraints attached on the traffic to be transferred. An alternative to handling individual flows is to have the network look at aggregates. In such a case, the IP packets have to be marked in order to assign them to the proper aggregate. Then, it may not be needed for the application to signal its request to the network in advance; it could simply start sending packets. Thus, it would not be up to the network to report explicitly to the application if a service request cannot be met, it simply has to be detected by the application itself (e.g. as lost packets). This resembles DiffServ. The former approach resembles IntServ, allowing for a finer and closer following up of the network for each application and traffic flow initiated.

In order to enhance the support of differentiated service levels some issues are mentioned in [RFC2990]:

- Aware applications. In case the application is capable of giving estimates of its requests to the network, the network may check its state to see if the requests can be met. This can then be returned to the application. This requires that the application can give relative precise description of its traffic profile (and policing may then be applied). Then, the application may be made aware of which service level that is provided, e.g. in case different charging applies. Another factor is that making aware applications could allow for end-to-end views between applications at the two endpoints, ensuring that the receiver is prepared to receive the information to be transmitted. However, it can always be discussed which should be the first to be upgraded: the network or the applications. Preferably these should go hand-in-hand.

- Scalable and accurate service environment. As noted, IntServ allows fine granularity following traffic flows, resource allocation, and conveying information to the other end-point. The so-called scalability, however, may be a problem. On the other hand, DiffServ scales well, while not supporting signalling. Some effort is undertaken to enhance DiffServ with capabilities for signalling and reservation of resources.

- Service query and discovery. Prior to using a service, an application may need to decide if the service can be supported by the network. This could likely be used for initiating a negotiation between the application and the network.

- Service levels on resource handling and routing. Considering service levels when deciding upon routing and configuring resources requires that additional attributes are introduced. These attributes would describe the service levels, or characteristics of the resources, and have to be correlated with corresponding attributes on the service requests. In addition, features like load distribution could be achieved.

- Relations with TCP. Recognising that TCP is one of the most used transport protocols, having efficient means for controlling the TCP flow rate is essential. TCP relies on ACK messages in order to adjust its rate as part of the load control. When introducing different service levels, the TCP should get the proper feedback on the forward direction as this is the one to be controlled. Thus, effects in the backward direction should not have too much influence on the TCP estimates of required flow rate.

- Granularity of flow identification. As discussed earlier, IntServ and DiffServ apply different levels of granularity. IntServ may recognise individual flows given by the 5 tuple (IP source address, IP destination address, source port, destination port and protocol). At the border of a DiffServ domain a similar 5 tuple can be applied to map the flow into a class/aggregate. Use of various tunnelling, e.g. IPSec and fragmentation of transport packets into multiple IP packets may imply that this information is not present/ detectable in every IP packet of the flow.

- Classes of service levels. In principle, a large number of service levels could be present. For instance, even DiffServ has 64 classes as options, then these may even be implemented differently by different networks along a path, leading to a vast number of options. Harmonising the service levels may result in easier interconnection and service provision in multiprovider environments, although service levels could also be seen as a competitive factor. Service discovery as mentioned above becomes even more requested for such configurations. Between the different networks this could also ask for enhancing the set of routing protocol attributes, including some describing the service levels supported.

- Measuring service level and delivery. In selling a service level to a customer, it is central to have means in place in order to document that the service has been delivered as agreed. Another purpose of such measurements could be as a basis for admission control and routing.

- Accounting for service levels. In case various service levels are to be offered, corresponding range of tariffing levels should be used as well. A technical argument is that use of more network resources should be reflected in a higher tariff. Naturally, other arguments may go against such a conclusion. This technical argument points towards application of usage-based charging.

According to [RFC2990] the following aspects are included in an architecture for service level/ QoS level:

a) Control the network response such that it is consistent and predictable;

b) Control the network response such that the service level is provided as agreed:

c) Allow establishment of agreed service level in advance;

d) Control contention for network resources such that the appropriate service levels are achieved;

e) Control contention for network resources such that a fair allocation is achieved (although fair has not been defined);

f) Allow for efficient utilisation of network resources while providing a range of service levels.

All these issues have to be addressed in order to ensure that the service levels can be provided. Actually, all these have to be in place in a coherent way to offer end-to-end service levels in agreed ways.

In addition to the issues listed above, more unanswered questions are found for interdomain configurations, like how to efficiently handle the set of SLAs in a multi-service and multi-provider configuration.

6.3 Overview of Further Challenges

In addition to the issues discussed above, more aspects can be looked at. The following description of the open issues is centred around the groups depicted in Figure 9:

- The network nodes sphere representing various nodes involved for IP transport, e.g. routers and hosts/terminals. Various segments of an operator's network, including access and core, are part of this. User terminals and relevant parts of applications can also be included. Typical issues are related to ensuring and monitoring performance of traffic flows, configuring resources, and so forth.

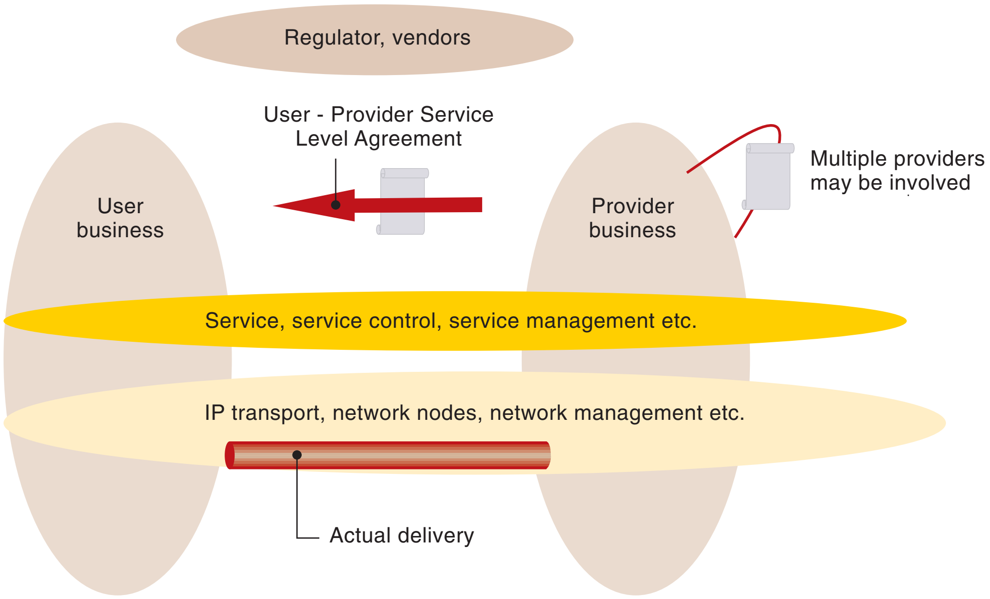

Regulator, vendors
User - Provider Service Level Agreement
Multiple providers may be involved
User business
Provider business
Service, service control, service management etc.
IP transport, network nodes, network management etc.
Actual delivery

Figure 9 A grouping of challenges related to ensuring QoS for IP-based services

- The service concerns, involving control and management. Functionality both in the user's equipment and in the provider's domain is included. Typical issues are definition of service and corresponding QoS (e.g. which parameters to apply), integration of control and management, elaboration of SLAs (considering multi-provider and multi-service) – between providers and towards end-users, applications capable of expressing their service level demands, and so forth.

- The business concerns, addressing processes and (internal) models within an actor (e.g. provider or user). Typical issues are description of internal processes for providing/delivering IP services, processes for collecting and storing performance data from different sources, methods for searching optimal arrangements for delivering services.

Traffic Engineering should be aware of these issues, incorporating appropriate procedures and interfaces.

Naturally, the above-mentioned groups are just giving one, non-exhaustive, way of dividing the various issues. A further complicating fact is that quite a few of the various issues are inter-dependent, not making it easy to clearly separate the questions to be addressed.

6.3.1 IP Transport Concerns

Several open issues for QoS related to transport of IP packets are described in [RFC2990]. Although these were discussed above, they, as well as others, are summarised in the following:

- Monitoring capabilities. How to monitor the traffic flows, resource utilisation and QoS-related performance in an efficient way? This includes decision on what level of granularity to look at, that is, what aggregate of flows, and time-scale to apply. Furthermore, the monitoring results must be applicable to document the conditions stated in SLAs (which well may refer to "higher-level" services). Monitoring results are also used for further tuning of resource configuration and traffic handling.

- Configuring resources. TE mechanisms are needed to properly/efficiently configure the network resources and control the traffic load such that the SLA conditions are met. This involves routing, allocation of resources (e.g. bandwidth), admission control, and so forth. Considering the multi-service network, this is a rather complex matter where scalability might become a particular challenge. Particular attention is placed on configuring resources for carrying voice-over-IP (also likely to involve gateways).

- Completion of appropriate standards. At least on a shorter term combinations of DiffServ and MPLS are promoted for use in the core network. Hence, completion of standards within that area is needed, in particular to simplify interoperability between domains. Regarding access, IntServ may also be applied. Then, solutions are needed for this as well, in addition to mapping between IntServ and DiffServ/MPLS.

Establishing an efficient IP-based network also requests interworking with layers below and above IP. For example, co-ordination of functions in the optical layer and the IP layer should be utilised. The same goes for applications and other "higher layer" functions, like policy and directory.

6.3.2 Service Concerns

On quite a few occasions it is seen that specifying the actual service to be delivered is not done adequately. Hence, there may be some room for interpretation both by the provider and by the user. As more higher-level services (i.e. above mere transport of IP packets) are "sold", the TE mechanisms should capture even these aspects. This makes the scope somewhat broader than what mainly springs to mind. Some essential issues in this group are:

- SLA/SLS/TCA/TCS. Elaborating agreements and specifications at the proper level of detail which are accurate, is still a challenge. Designing SLAs in a multi-provider/multi-service environment raises additional challenges, like how to relate two SLAs with independent providers referring to the same access line/user. Based on different events occurring in the network, different users may be affected. Ways of identifying service affecting faults and which users that are bothered will then be requested. Particular challenges may arise from avoiding scalability problems.

- Management models. Both service management and network management have to be completed for efficient management. Regarding network management, issues like collecting monitoring data, configuring network resources (e.g. for MPLS paths) have to be implemented. Regarding service management, there are some suggestions for the need for more centralised management architecture, supported by stronger admit and control mechanisms of flows at the edge of the core network [ID_IPsm]. Then, the end-to-end (including multi-domain) view should be taken on by service management, compared to only looking at portions of it, as is most common today. Policy-based management might well be applied, also accommodating the dynamics of the network and the usage/traffic flows.

- Integrated control and management. To some extent control and management procedures may perform similar tasks, like establishing an MPLS path. Therefore, figuring out efficient ways of combining control and management is needed.

- Applications and service discovery. Applications must be upgraded to be aware of the services and service levels offered by a network. Functions for discovery and negotiation should therefore be present, both in the terminals/applications and in the network. In addition, applications should provide estimates of their requests from the network, like estimates of the traffic flow characteristics (traffic profile).

- Controlling traffic flows. In several of today's IP-based networks, TCP is used for flow control. This should also work properly even when several service levels are introduced. In addition, controlling the traffic flow in other ways may be possible. One example is using dynamic charging schemes, which is investigated although not concluded on.

- Completing standards. For efficient implementation of control and management regimes, standards are crucial. Appropriate standards, e.g. for RSVP, policy and SLA management, are needed.

6.3.3 Business Concerns

Deciding upon QoS and internal arrangements for handling traffic are parts in the business activities. Moreover, when settling the appropriate value parameters and utilisation of mechanisms, one would likely face several trade-offs, like what level to offer at what price, which mechanisms to implement within its domain, and so forth. Similar trade-offs have already been parts of the business decision process. Therefore these aspects may smoothly fit into those concerns.

Carrying out business-related evaluations, the market situation is taken into account. Potential customers' requests, competitors' activities, regulatory directives, etc. would be considered. Bearing in mind that service degradations/failures may occur, stating conditions in the agreements can be looked upon as risk taking. That is, damages/penalties in case an event happens are balanced against the cost of the means undertaken in order to lower the probability of the event occurring. Lower cost is commonly sought, while major negative consequences are avoided. Balancing internal mechanisms and the conditions stated in agreements towards any sub-providers would also be part of this picture as seen by a provider.

A few issues related to business are listed in the following:

- Processes and data flows. An adequate model of processes and flows related to a provider is needed to implement systems allowing fast, accurate and automatic service provision/delivery. Considering that several sources and databases may be involved, keeping consistencies and collecting relevant data may be a challenge, in particular when the data bases are managed by different providers.

- Cost – revenue analysis of related mechanisms. There still seems to be two overall suggestions to the QoS challenges: i) introducing more QoS-related mechanisms; or, ii) introducing more capacity, keeping the network/system simple. An analysis of this could be conducted, estimating any real benefit from having ensured and differentiated IP-based services.

- Optimising services, resources and SLA conditions. What service classes to offer, how to deploy the network resources optimally, how to state and balance SLA terms towards users and secondary providers against the need for own resources, are a few questions that need to be answered by a provider. Hence, methods to assist a provider in searching for the answers are needed. A user does not commonly specify their quality needs merely from a communication point of view, but rather from the consequences they envisage on their business and human relations – the secondary scope – if a failure occurs. This also determines the level of QoS they are willing to pay for regarding a specific service ordered.

- Then, guidelines to a way of assisting the customer in structuring the consequences that they might envisage – both in primary and secondary scope – would be requested.

- Accounting, charging and billing. In introducing a set of service classes, tariffing schemes have to be defined as well. Considering interconnections, schemes and mechanisms for exchanging accounting data is also necessary.

- Communicating with the human end-user. Of particular interest are the QoS parameters that communicate well and unambiguously to the non-professional users – i.e. residential end-users. Imperative to all parameters is that they are harmonised and allow for mapping into others. Monitoring a QoS parameter could be done both “continuously” and by “sampling” according to “typical” usage. The dynamics of the QoS may have major influence on how QoS is perceived.

7 Concluding Remarks

One intention is that introducing “sophisticated” mechanisms related to QoS, and TE in general, allows the provision of a wider portfolio of services and accompanying QoS levels. At the same time the network resources are efficiently utilised. These mechanisms introduce a cost in the form of increased overhead/complexity, meaning that the basic question is that such cost must be weighed against the benefits that can be obtained.

The quality-efficiency product has been proposed by [Bern00], see Figure 10, showing the trade-off between network utilisation and service provision guarantees. The illustration gives that higher loads on the resources, e.g. links, can be used if less strict values of the QoS are given. However, introducing more QoS-related mechanisms may allow for higher levels of resource utilisation for the same level of guarantee. Hence, if an operator wants to operate a network efficiently while still supporting strict guarantees, more sophisticated QoS-related mechanisms must be introduced. The different QoS-related mechanisms may imply different levels of overhead in terms of processing and storing.

The quality-efficiency product is valid for a certain network domain. In an end-to-end view, such a product/measure may not have the same value for all domains. That is, some domains may have high utilisation, while for others a low utilisation is allowed. Naturally, there is no clear boundary between high and low utilisation, as between high and low levels of guarantee.

Returning to the issue of demand, not all traffic flows (and users/applications) will ask for strict guarantees. One question is whether to define a number of virtual networks, e.g. with different quality-efficiency products, on the same physical network. This would then open for a broader spectre of service levels, better matching the different customer groups.

In this paper, the basic activities and mechanisms of TE have been described. A motivation is to provide an introduction to the topics. Many of the issues are treated in accompanying papers in this issue of *Telektronikk*.

Although there are quite a few results available, essential issues also remain to fully support the multi-service and multi-provider configuration. A few of these are briefly described above. Hence, the continuing need for improving Traffic Engineering solutions in IP-based networks should be beyond doubt.

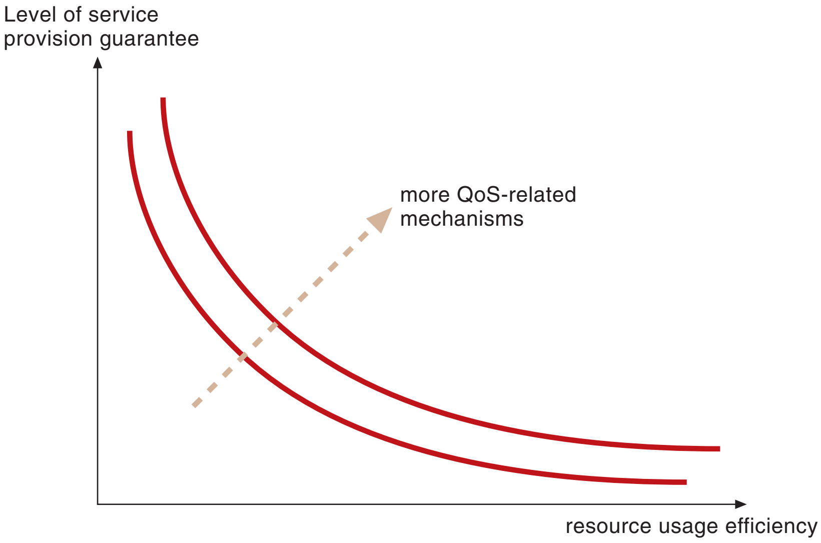


Figure 10 Quality – efficiency product, from [Bern00]

References

[Bern00] Bernet, Y. 2000. The Role of the Host Supporting the Full Service QoS Enabled Network. In: *Internet2 workshop*. Houston. URL: http://www.internet2.edu/qos/houston2000/proceedings/Bernet/20000210-QoS2000-Bernet.pdf

[Come88] Comer, D. 1988. *Internetworking with TCP/IP*. Englewood Cliffs, NJ, Prentice-Hall.

[ID_IPsm] Eder, M, Chaskar, H, Nag, S. *IP Service Management in the QoS Network*. draft-irtf-smrg-ipsm-00.txt. July 2001. Work in progress.

[ID_resreq] Kirstaedter, A, Autenrieth, A. *An Extended QoS Architecture Supporting Differentiated Resilience Requirements of IP Services*. draft-kirstaedter-extqosarch-00.txt. July, 2000. Work in progress.

[ID_tepri] Awduche, D O et al. *Overview and Principles of Internet Traffic Engineering*. draft-ietf-tewg-principles-00.txt. Aug. 2001. Work in progress.

[Jens01] Jensen, T. Internet Protocol and Transport Protocols. *Telektronikk*, 97 (2/3), 20–38, 2001. (This issue.)

[RFC2990] IETF. 2000. Huston, G. *Next Steps for the IP QoS Architecture*. (RFC 2990.)

Basic IP-related Mechanisms

TERJE JENSEN

When reading about IP-based networks, one almost immediately bangs into a series of abbreviations and expressions. The main objective of this paper is to outline the basic mechanisms; Best Effort, Differentiated Services, Integrated Services, MultiProtocol Label Switching, Resource reSerVation Protocol, Label Distribution Protocol, and some of the ways packets are buffered and scheduled in a router.

Terje Jensen (39) is Research Manager at Telenor R&D, Kjeller. He earned his PhD degree in 1995 from the Norwegian University of Science and Technology. Activities include performance modelling and analysis, dimensioning and network evolution studies.

terje.jensen1@telenor.com

1 Introduction

From the outset, one may say the Internet Protocol (IP) was intended to transport information from a source to a destination. The main emphasis may be that the information eventually arrived with less strict requirements on time. This may work well for some applications, like computers and sensors exchanging data, and when no impatient human being is involved. However, as more types of applications are loaded on the IP-based networks, additional requirements are also placed on these networks. Hence, the question arose how to efficiently support these applications. This became even stronger as the commercial concerns grew for providing services on the IP-based networks.

This may be the main motivation for proposing the mechanisms described in this paper. Moving beyond the best effort, other service models were introduced; the Integrated Services (IntServ) and the Differentiated Services (DiffServ) models. Protocols for reserving resources were also proposed, including the Resource reSerVation Protocol (RSVP). Avoiding the routing processing and introducing means for load distribution and traffic flow protection, the MultiProtocol Label Switching (MPLS) was described. These are described in the following chapters. Some basic packet handling mechanisms are outlined first (Chapter 2).

Quite a lot of variations and detailed information is available on these subjects, and references are pointing at the sources. In addition to the huge background material refinements are steadily on-going.

2 Congestion and Packet Handling

Congestion is said to arise when too many packets (too high a load) are present in a sub-network compared to the capacity of that sub-network. Congestion has a tendency to feed upon itself, for example as packets queued up may be timed out and retransmitted adding to the congestion. Another phenomenon is the spreading as unacknowledged packets will be buffered and do not release memory space.

In this chapter means for managing congestion are described. Basically, they are addressing how packets are handled in an end-system and in a router. The flow/congestion control mechanisms in TCP are essential, as described in [Jens01].

Congestion avoidance is simply to avoid the occurrence of congestion. Congestion avoidance constitutes preventive congestion management policies that can be categorised as having a long, medium or short response time scale. Long-term policies include capacity planning to expand network capacity using estimates or forecasts of future traffic demand and distribution. Medium-term policies cover the minutes to days time scale. Examples are adjusting routing parameters and reconfiguring the logical network topology. Short-term congestion avoidance includes packet level processing functions as returned to later in this chapter.

Congestion control is commonly used to make sure that a sub-network is able to carry the offered traffic efficiently, involving all traffic flows passing. On the other hand, *flow control* is used between a pair of sender and receiver preventing the sender from transmitting packets too fast, involving feedback from the receiver to the sender. However, such a feedback may also be set by the network, e.g. to avoid congestion, implying that flow control mechanisms can be utilised related to congestion control.

Several mechanisms (called policies in [Tane96]) affect congestion and how it could be dealt with, see Figure 1, including:

- At the *end-to-end level*: retransmission policy, out-of-order buffering policy, acknowledgement policy, flow control policy, timeout policy.

- At the *network level*: use of connections, policy for queueing and serving packets, discard policy (buffer is full), routing policy, packet lifetime policy.

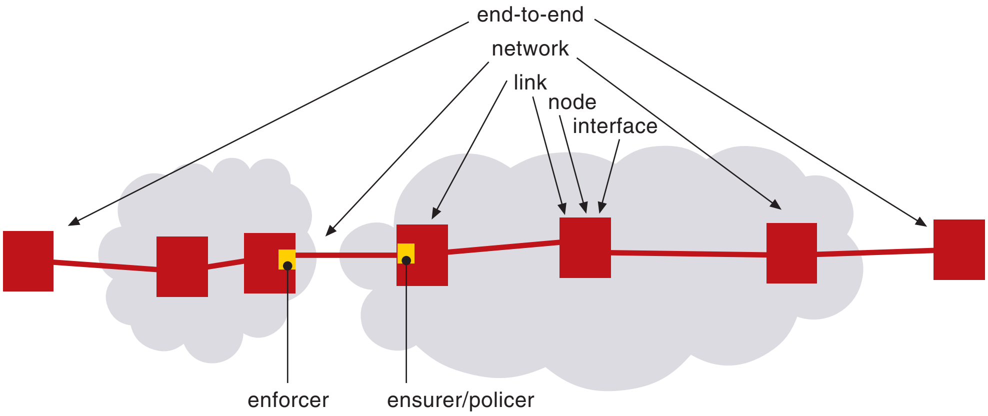


Figure 1 Levels/scopes for congestion control mechanisms

- At the *link level*: retransmission policy, out-of-order buffering policy, acknowledgement policy, flow control policy.

As seen from the list, policies for how the information is carried in an accurate manner affect the congestion management, as should be expected. In particular, appropriate means should be taken to avoid that an emerging congestion situation is further worsened, like simply to postpone the re-transmission of packets when the buffers are filled.

A basic cause of congestion is that the current traffic load is greater than the service capacity. Commonly, this would be due to failures in the service capacity (e.g. link failure) or that the traffic is in a burst. The latter can be managed by shaping the traffic, meaning that traffic peaks are made smaller. This might be observed at the egress (or supported at the ingress border), in fulfilment with the agreed traffic pattern. At the egress side this may be called a *traffic enforcer*. On the other side of the border, the traffic flow is frequently monitored and steps taken in case the expected behaviour is not followed. Then a *traffic policer* can be implemented, ensuring that the resulting load is as expected. Leaky buckets are frequently used for shaping and policing. When variable packet lengths are used, like for IP, the bucket may count in data volume, e.g. bytes, rather than in number of packets. The leaky bucket may operate on the data flow itself or on tokens (being permissions to send data). Suitable reactions must also be specified when too much traffic is arriving. One reaction is to drop the packet, another reaction is to change the class/priority of the packet.

In order to set up a shaper and policer a proper traffic flow specification has to be made. Such a specification may use parameters describing the traffic flow itself (mean bitrate, peak bitrate, burst duration, etc.) or parameters more related to a leaky bucket implementation (bucket size, leak rate; see following section). When such a specification has been done it may also be used for admission control, i.e. to decide whether or not more traffic should be admitted to the network.

Buffering schemes and queueing disciplines are important components in congestion handling. In particular, when a number of service classes is defined, those mechanisms have to maintain the differentiation between the classes, e.g. by serving orders, packet dropping levels, and so forth. Differentiations can be implemented for all mechanisms described in the following sections.

2.1 Policing

Policing is a general term used for the process of preventing a traffic flow from grabbing more resources than allowed. Actions to be taken on non-conforming traffic (packets) include dropping the packet or remarking the packet. Remarking can be utilised for putting the packet into another class/queue or to increase its probability for being dropped later in the network.

A *leaky bucket* algorithm is frequently used to define the conformance of a traffic flow. The

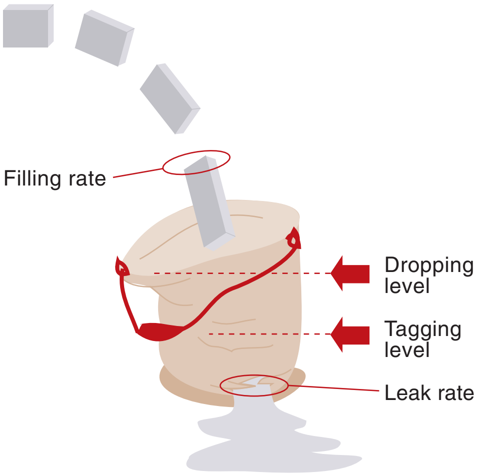


Figure 2 Analogy to leaky bucket

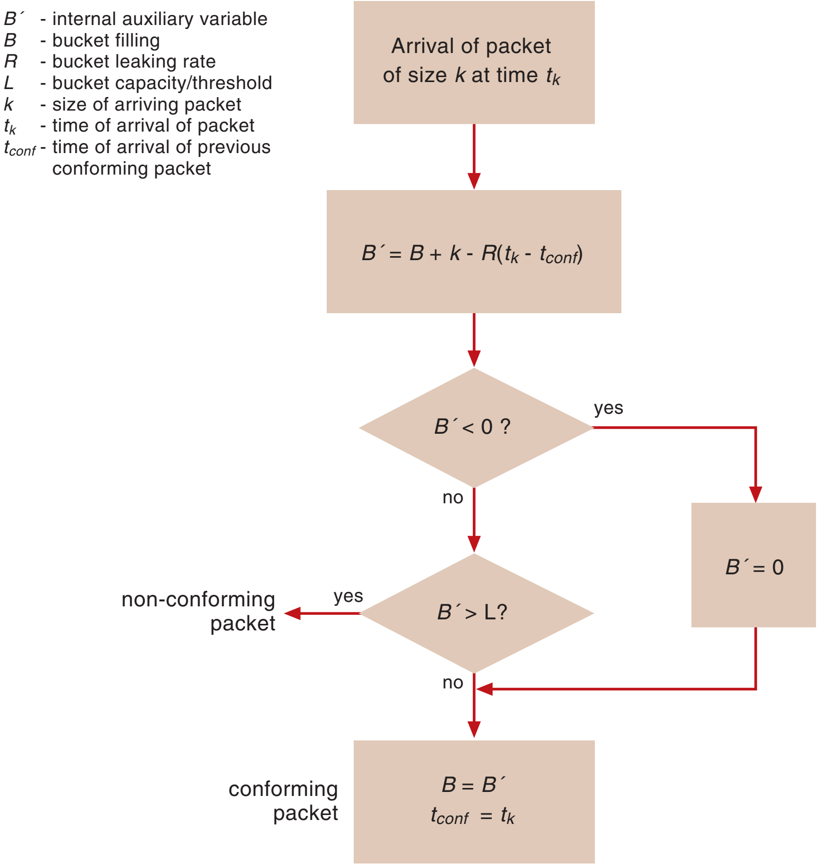

B' - internal auxiliary variable
B - bucket filling
R - bucket leaking rate
L - bucket capacity/threshold
k - size of arriving packet
t_k - time of arrival of packet
t_{conf} - time of arrival of previous conforming packet
Arrival of packet of size k at time t_k
$B' = B + k - R(t_k - t_{conf})$
$B' < 0$?
yes
$B' = 0$
no
$B' > L$?
yes
non-conforming packet
no
conforming packet
$B = B'$
$t_{conf} = t_k$

Figure 3 Illustration of leaky bucket algorithm

leaky bucket analogy refers to a bucket with a hole in the bottom that causes it to “leak” at a certain rate. Then more “fluid” is added to the bucket according to arrival of packets, see Figure 2.

Describing the arriving traffic/packets, several parameters can be used, like the peak rate and the mean rate. Therefore, the leaky bucket can also operate on different time scales according to the parameter that is checked. The “depth”/ capacity of the bucket corresponds to a tolerance for the parameter that is checked.

In principle, several levels of the bucket capacity can be defined, each level corresponding to a certain action, like re-marking packets or dropping packets.

In the algorithm, a counter represents the content of the bucket. This counter is incremented according to the size of the packet that arrives. The “leak rate” in the algorithm is the decrement rate, which reduces the counter value by a given value at certain intervals. The bucket volume is analogous to the counter range, which is represented by the permissible time tolerance for the incoming cells. An algorithm flow is shown in Figure 3.

As mentioned above, several time scales/parameters of the traffic flow would typically be operated on. In addition, several classes may also be given for the traffic flow (i.e. a flow aggregate). Above the bucket is described for the data flow itself, however, the bucket may also refer to an amount of tokens as described in the following.

Policing devices applying the leaky bucket algorithm are described in [RFC2697] and [RFC2698]. These describe the single rate Three Colour Marker (srTCM) and two rate Three Colour Marker (trTCM), respectively. As several classes may be assumed, these would be applicable to DiffServ classes, in particular the Assured Forwarding class (see Chapter 4).

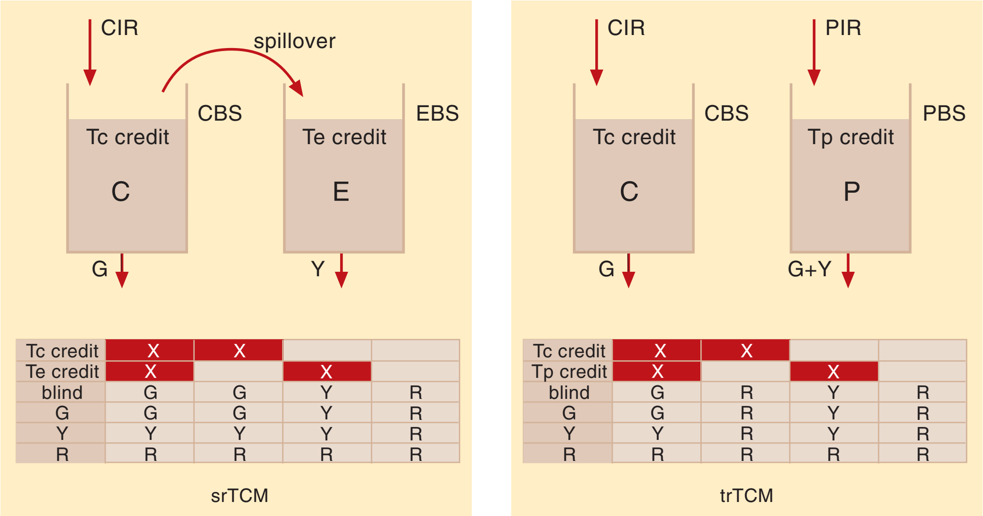
The srTCM meters a traffic flow and marks packets according to three traffic parameters; Committed Information Rate (CIR), Committed Burst Size (CBS) and Excess Burst Size (EBS). The marking may be green, yellow or red. A packet is marked green if it does not exceed the CBS, yellow if it does exceed the CBS and not the EBS, and red otherwise.

The trTCM meters a traffic flow and marks packets based on two rates; Peak Information Rate (PIR) and CIR, and their associated burst sizes. Red is used if the packet exceeds the PIR. If not, it is marked yellow if it exceeds the CIR, and green otherwise.

These meters may operate in two modes; either colour-aware or colour-blind. In the former it is assumed that the packet has already been marked when arriving at the meter. When colour-blind, the meter assumes that no colour has been attached to the packet. The colour is coded into the DiffServ field (see Chapter 4).

The behaviour of the srTCM meter can be modelled as two token buckets, *C* and *E*, which both share the common rate CIR, see Figure 4. The maximum size of the token bucket *C* is CBS and the maximum size of the token bucket *E* is EBS. Tokens are generated at a rate equal to CIR and inserted into token bucket *C*. If token bucket *C* is full, tokens spill over to token bucket *E*. If also token bucket *E* is full, the token is discarded.

When the srTCM is configured in colour-blind mode, it treats all received packets as unmarked packets. The colour of the packet is determined by the status of the token buckets upon its arrival. A packet of size *B* bytes is marked as green if token bucket *C* contains at least *B* tokens upon the packet’s arrival. If this is not the case, it is marked as yellow if token bucket *E* contains at least *B* tokens. If none of the token buckets contain at least *B* tokens, the packet is marked red. When the decision is made to mark the packet as green or yellow, *B* tokens are removed from the associated token bucket.



Tc credit	X	X		
Te credit	X		X	
blind	G	G	Y	R
G	G	G	Y	R
Y	Y	Y	Y	R
R	R	R	R	R

srTCM

Tc credit	X	X		
Tp credit	X		X	
blind	G	R	Y	R
G	G	R	Y	R
Y	Y	R	Y	R
R	R	R	R	R

trTCM

Figure 4 Single rate (left) and two rate (right) Three Colour Marking

When the srTCM operates in colour-aware mode, arriving packets are considered to be pre-marked. Then, the marking must be more conservative in the colouring of packets. That is, the re-colouring is only allowed if it results in a higher drop probability (change green to yellow/red or change yellow to red) for the packet. The algorithm described above is followed when a green packet arrives. When an arriving packet is pre-coloured as yellow, only the status of the *E* token bucket is considered. If the packet has a size of *B* bytes, the packet remains yellow if token bucket *E* contains at least *B* tokens upon its arrival. Otherwise, it is re-coloured as red. A packet pre-coloured as red remains red.

Two token buckets can be used also for modelling the trTCM (Figure 4). Tokens are added to the token buckets at rates CIR for *C* and PIR for *P*. In colour-blind mode, a packet of size *B* bytes is coloured as red if token bucket *P* contains less than *B* tokens upon its arrival. If token bucket *P* contains at least *B* tokens, it is checked whether token bucket *C* also contains *B* tokens. If it does, the packet is coloured as green and *B* tokens are removed from both buckets. Otherwise, it is coloured as yellow and *B* tokens are removed only from token bucket *P*. The colour-aware mode of operation is similar to the above description.

As mentioned above the TCM policing is frequently carried out at the boundary of a DiffServ domain. Boundary nodes could limit traffic carried on behalf of customers to the constraints specified in the associated Traffic Conditioning Specifications (see Chapter 4).

2.2 Buffer Management

The basic buffer management scheme is to treat all packets equally; insert the packets into a queue upon arrival and take them out of the queue for transmission on a link. The buffer management schemes may be operating on different aggregates of traffic flows, e.g. from all packets on an interface to individual traffic flows.

Applying Tail-Drop (TD) arriving packets are dropped only when the queue is full. A problem with tail drop is that global synchronisation of TCP sources can occur as multiple TCP sources reduce their transfer rates almost at the same time. Then congestion clears and the TCP sources gradually increase their transmission rates again until a new congestion situation may build up. This could result in longer periods during which the transmission link is not fully utilised. That is, oscillations of the link load could be observed with a poor average utilisation.

2.2.1 Random Early Detections and Derivatives

Active queue management mechanisms can be designed to avoid the synchronisation of TCP connections. A goal of such an active queue management mechanism is to make each TCP connection reduce its sending rate at different moments. An essential algorithm for this is the Random Early Detection (RED). When applying this algorithm packets are discarded randomly when the buffer is near to congestion. In this way, TCP connections will lose packets at different time instants, avoiding the synchronisation between TCP connections. RED supports congestion avoidance by controlling the average queue size. During congestion (but before the queue is filled), the RED scheme marks arriving packets according to a probabilistic algorithm which takes into account the average queue size. The marked packets can be dropped as an early congestion notification before queues actually overflow. This will trigger corresponding TCP

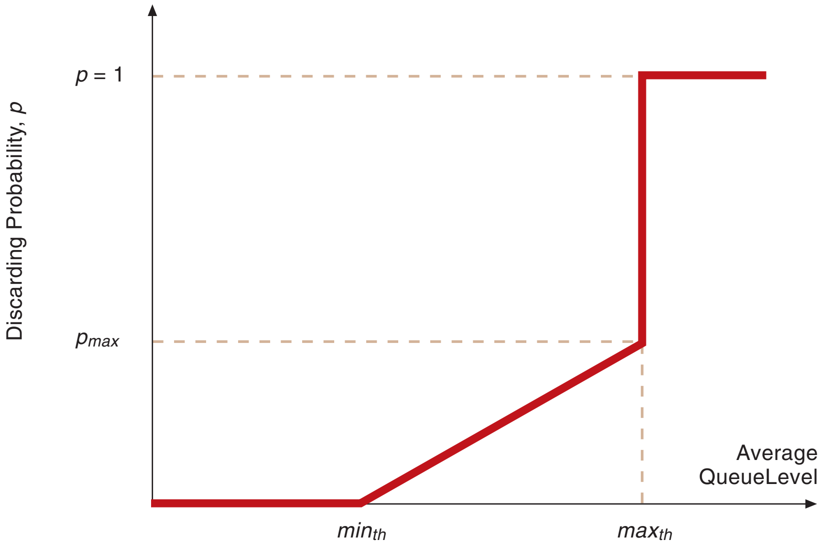


Figure 5 RED algorithm

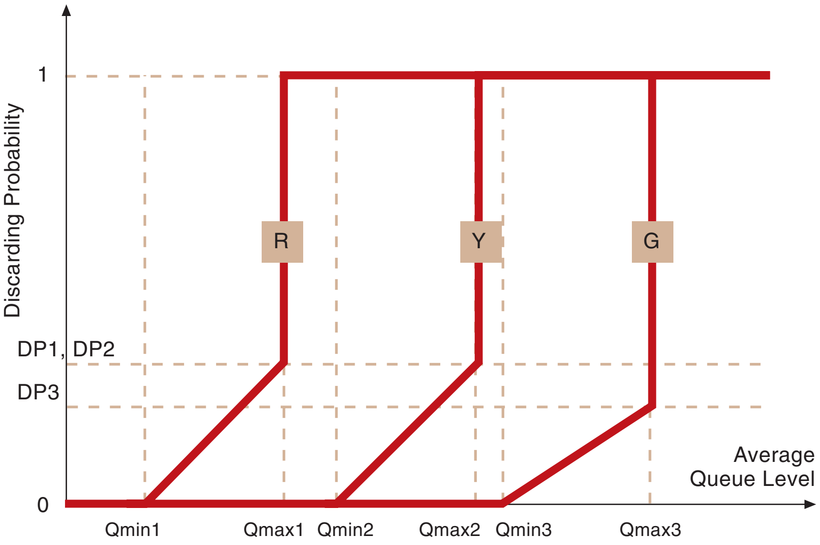


Figure 6 WRED with three classes (red, yellow, green)

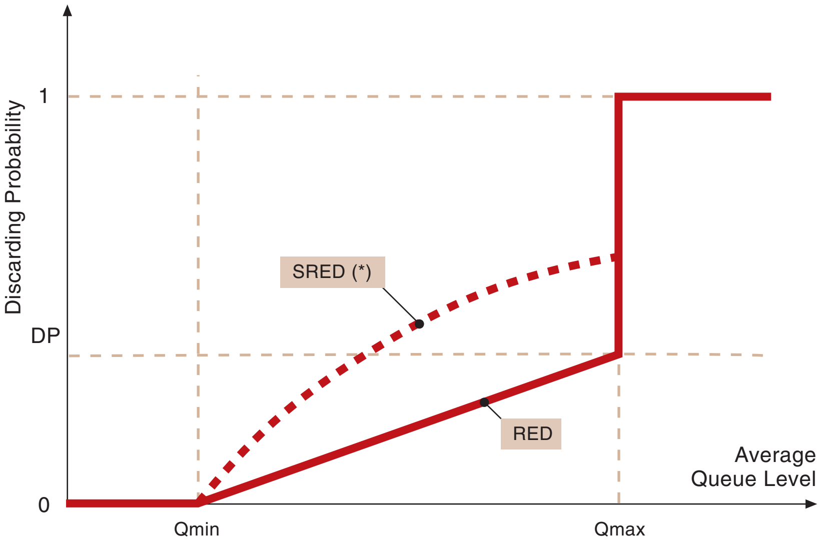


Figure 7 Illustration of Shock-absorber RED compared to basic RED

sources to slow down. RED is mainly useful when the bulk of the traffic is TCP traffic. A potential consequence of RED is that UDP sources, or some misbehaved greedy sources can obtain an unfair advantage when TCP connections slow down their rate.

The RED algorithm works as follows (ref. Figure 5): When a packet arrives, the algorithm calculates the average queue size, for instance using a low-pass filter with an exponential weighted moving average:

- If the average queue size is lower than a minimum threshold (min_{th}), the packet is queued.
- If the average queue size is greater than a maximum threshold (max_{th}), the packet is discarded.
- If the average queue size is greater than the minimum threshold, and lower than the maximum one, the packet is discarded with a probability *p*, which is a function of the average queue size.

Weighted RED (WRED) is a RED-derived mechanism that assigns to each class a different RED algorithm. Then it is possible to differentiate between the different classes. Basically WRED provides RED with separate thresholds and weights for different classes. For example, standard traffic may be dropped more frequently than premium traffic during periods of congestion.

As an example classes like green, yellow and red may be used (as for the policing algorithm). This is illustrated in Figure 6. Another example is to use different dropping probabilities for TCP and for UDP traffic flows.

RED and WRED commonly use the average queue sizes, not considering the link utilisation, which again could lead to oscillations for the queueing level under congestion. To address this a Shock-absorber RED (SRED) has been derived, see Figure 7. Then the instantaneous dropping probability depends not only on the queue size but also on the offered load. This is to reduce the variations for the queue filling. SRED may also be extended to allow for several traffic classes.

RED with In/Out bit (RIO) is a RED-derived mechanism that assigns two different priorities. But instead of using the same average queue size for both priorities (like WRED does for all the classes), it uses the average queue size for OUT (out of profile) packets, and the average queue size without taking into account the queued OUT packets for IN (in profile) packets, see Figure 8.

Some experiments show that RIO may offer better results than WRED when parameters are correctly set. However, a number of additional parameters are to be given in RIO, such as those of IN priority. These are influenced by the given scenario. Therefore, WRED may be preferred by some when robustness is required.

Estimating the better combinations may well be a challenge in itself, considering the number of parameters that may be asked for and the dynamics in the traffic flows.

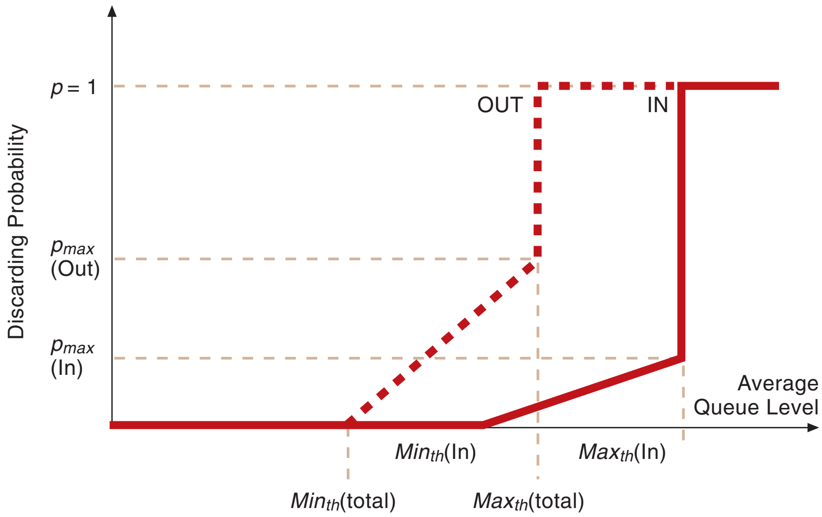


Figure 8 RIO algorithm

2.2.2 Explicit Congestion Notification

Explicit Congestion Notification (ECN) has been suggested as one way of tackling congestion handling. Two bits in the IP header are used: Congestion experienced (CE) – bit 7 of the ToS field, and ECN capable transport (ECT) – bit 6 of the ToS field. For IPv6 the corresponding bits in the traffic class field are used. The ECT bit is set by a sender able to react on indication of congestion by ECN.

In addition TCP has to be modified in order to carry the information back to the sender. TCP can be said to treat the network as a black box, only reacting on lost packets without any further insight into the situation in the network. This might lead to low utilisation of the network. Therefore, active queue management, ref. [RFC 2309] has been promoted to avoid some of the drawbacks of packet dropping when queues are (almost) full. Random Early Detection (RED) is one example of an active queue management mechanism.

The ordinary TCP algorithm will not assist applications that are sensitive on delays (and packets arriving too late). Some algorithms, like ECN, could be introduced to allow the source to adjust its behaviour without experiencing too low throughput (due to packet loss) or long delays. Then the active queue management could rather set the CE bit than drop the packet when congestion is building up. This allows the receiver to get the packet, avoiding retransmission, at the same time as a congestion indication is conveyed. However, for IPv4 the header checksum has to be updated when the CE bit is set. This can be done incrementally as described in [ID_ecn].

When a sender gets information on congestion by ECN, it is supposed to behave as if a packet was dropped. One argument for this is to provide fairness compared to non-ECN systems. Thus a router, in case a congestion threshold is exceeded, may drop a packet when the ECT bit is not set and set the CE bit in packets where the ECT bit is set.

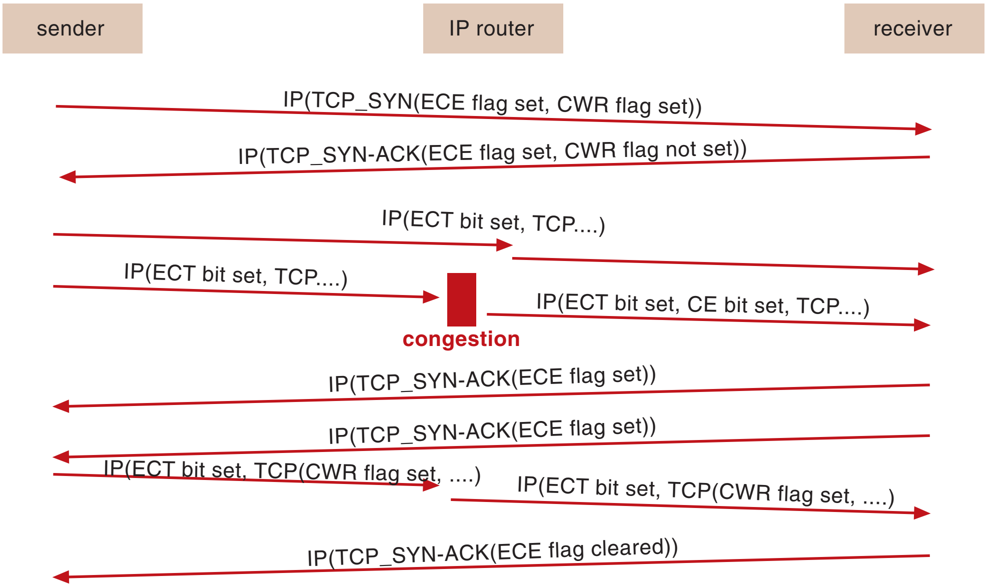

sender
IP router
receiver
IP(TCP_SYN(ECE flag set, CWR flag set))
IP(TCP_SYN-ACK(ECE flag set, CWR flag not set))
IP(ECT bit set, TCP....)
IP(ECT bit set, TCP....)
congestion
IP(ECT bit set, CE bit set, TCP....)
IP(TCP_SYN-ACK(ECE flag set))
IP(TCP_SYN-ACK(ECE flag set))
IP(ECT bit set, TCP(CWR flag set, ...)
IP(ECT bit set, TCP(CWR flag set, ...)
IP(TCP_SYN-ACK(ECE flag cleared))

Figure 9 Illustration of ECN-related messages

In order to make use of the ECN, support from the transport protocol is needed. For TCP three additional functions are identified: i) negotiation between the end-points during connection establishment to decide whether or not they are ECN-capable; ii) an ECN echo flag in the TCP header informing the sender that a CE-marked packet has been received (bit 9 in the Reserved field of the TCP header); and iii) a Congestion window reduced flag in the TCP header allowing the sender to inform the receiver that the congestion window has been reduced (bit 8 in the Reserved field of the TCP header).

The sequence of messages related to ECN is illustrated in Figure 9.

TCP should not react on congestion indications more than once every congestion/acknowledgement window (or about once every round-trip time). That is, a sender should not reduce its congestion window more than once in response to a message dropped and/or packets having the CE bit set out of a single window. It is stated that the ECT bit should not be set for pure TCP-ACK messages as no flow control is related to such messages. The same goes for TCP window probing, that is for packets generated periodically by the sender when the receiver has announced a zero window. As explained in [ID_tcpecn] nor should the ECT bit be set on retransmitted packets. Furthermore, the receiver should ignore the ECN field on the out-of-window data packets. Both these means are introduced to increase the security against denial-of-service attacks, i.e. where an attacker may spoof the IP source address and send packets making the receiver asking the real sender to decrease its sending rate.

For the moment, no special concerns are given when ECN is used within MPLS or other layer 2 transport means. For other ways of tunnelling, i.e. IP in IP, two options have been described:

- Full-functionality where the ECT bit of the inner header is copied to the outer header. At decapsulation, if the ECT bit of the inner header is set, the CE bit of the outer header is ORed with the CE bit of the inner header to update the CE bit of the inner header.

- Limited-functionality when ECN is not used for the IP tunnel. This is done by turning off the ECT bit of the outer header and not altering the inner header.

2.3 Scheduling

When implementing a network offering different traffic classes, queueing and scheduling mechanisms have to be configured. The choice of the better mechanism to use to obtain the declared service differentiation is far from obvious. This is one of the reasons for vendors to implement a variety of different queuing and scheduling mechanisms. As an example a few of the different mechanisms found in the Cisco routers are:

- Modified Deficit Round Robin (MDRR): MDRR may be able to provide special support for delay sensitive traffic, such as Voice-over-IP. MDRR includes a low-latency, high-priority queue that is treated differently from the other queues. This special queue is used to handle delay-sensitive traffic. It is possible to configure MDRR for strict priority handling of this queue. When that queue contains packets, it is served first until all of its packets are sent. Within MDRR, IP packets are mapped to different class-of-service queues, e.g. based on precedence bits (part of the ToS field). The remaining queues are served in a round-robin fashion.

- Weighted Round Robin (WRR): WRR is a packet queuing and scheduling algorithm, which provides class differentiation, bandwidth allocation, and delay bounding features. Hence, it is possible to give voice packets premium service, although not strict priority.

- Weighted Fair Queuing (WFQ): WFQ is an algorithm that provides priority management, but not strict prioritisation, during periods of traffic congestion. WFQ offers a solution that provides consistent, fair response time, based on weights. Hence, WFQ supports features such as traffic isolation and delay bandwidth guarantees.

  A second type of WFQ called Distributed Weighted Fair Queuing (DWFQ) provides bandwidth allocations and delay bounds to specified traffic flows by segregating the traffic into classes and then using first-in, first-out (FIFO) service to the various queues according to their assigned weights. There seems to be two kinds of standard WFQ (Flow-based WFQ, Class-based Weighted Fair Queuing, CBWFQ) and three kinds of DWFQ (Flow-based DWFQ, QoS Group-based DWFQ, ToS-based DWFQ) implemented in the Cisco routers.

- Internet Protocol Real-Time Transport Protocol Priority (IP RTP Priority): With the IP RTP Priority feature it is possible to specify a range of UDP/RTP ports whose traffic flows receive strict priority service over any other queues or classes. Strict priority means that if packets exist in the priority queue, they are fetched from the queue and sent first.

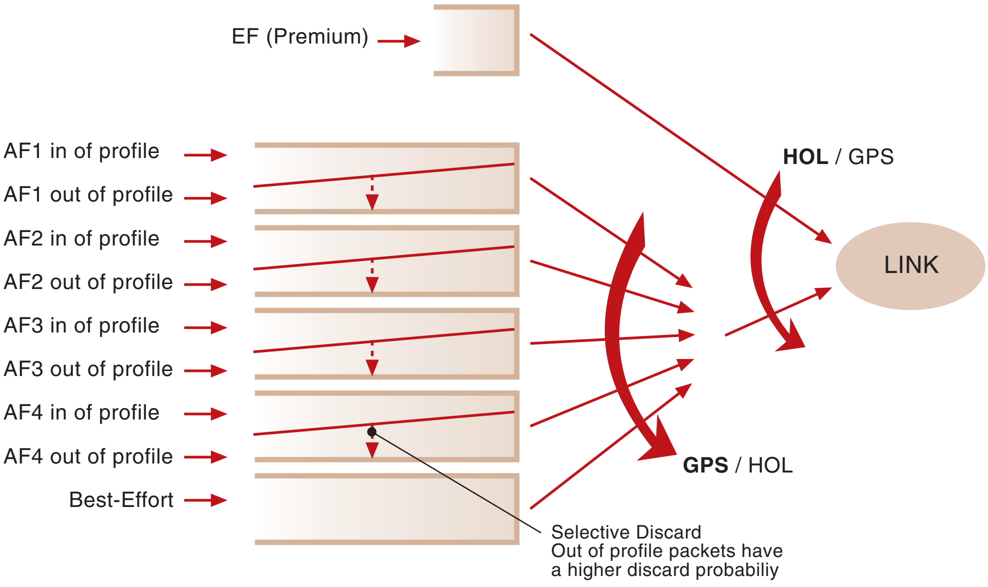


Figure 10 Example of outgoing link side in a router when DiffServ is implemented

- Priority Queuing within CBWFQ: The priority queuing within the CBWFQ feature brings the strict priority queuing functionality of IP RTP Priority required for delay-sensitive, real-time traffic, such as voice, to CBWFQ.

A schematic illustration of an outgoing interface in a router is depicted in Figure 10.

The different serving policies have their characteristics and corresponding preferred area of applicability. Normally, Head-Of-Line (HOL) is used such that real time traffic is given priority over elastic traffic. The problem is that this high-priority traffic would cause starvation for lower classes during high load. On the other hand, General Processor Sharing (GPS) disciplines (such as WFQ) are preferred if a minimum bandwidth must be guaranteed for each class. However, this kind of scheduling discipline is more complex to implement than HOL, and is susceptible to *priority inversion* if there is a higher-class congestion. That is, a lower class can actually get a better service if there is congestion in a higher class.

So, an observation may be that HOL is preferred if higher priority classes demand is much lower than lower classes demand. On the other hand, GPS may be preferred in some cases if higher priority classes demand is much higher than lower classes demand. Besides, admission control can be introduced to limit the load in each class to allow for bounds on the service levels.

When looking at some actual router implementations, one may find that several queueing and scheduling steps may be arranged in series. For example, on the output link, there may first be queues per service class on the IP level. Then, there may be queueing for placing packet flows into a transmission system, and, lastly, queueing for being transmitted on the link. For the last queue a single First-In-First-Out queue is often seen. Depending on the sizes and capacity of service, all these queues may impact the actual traffic flow characteristics, experienced delays, effective service differentiation, etc.

2.4 Admission Control

Quoted from [COST257] admission control is *a preventive traffic control which aims to admit an arriving new traffic source if and only if its quality of service as well as that of the already accepted sources is guaranteed. The admission control procedure should also ensure a high utilisation of network resources through efficient statistical multiplexing.*

Here a source may generate a set of flows. Remembering that *flow* may be defined as *a unidirectional succession of packets related to a certain (part of an) application. Packets belonging to the same flow have the same identifier (e.g. given by source and destination addresses and port numbers) and are initiated within a maximum separation in time between each other.*

As stated, when running an application a number of flows might result. An example is a multimedia application covering voice, video, file transfers, interaction control, etc. All these flows should be served in order for the application to be run in a satisfactory manner. Hence, the term session is introduced. A *session* is a *continuous period of activity during which a user generates a set of flows (elastic or streaming type).*

It should be noted that the session term is seen with varying interpretations, like an FTP session, HTTP session, and so forth. Usually, the admission control acts on the flow level, not taking into account effects on the session level. An argument may be that an application, in case a flow is not accepted, may retry the transfer and then leave that operation to the end-system/application. Hence, the network would not need to be enhanced with capabilities enabling grouping of flows into sessions. For some services and users, however, the service portfolio (and the conditions stated in the Service Level Agreement, SLA) may refer to phenomena on the session level. This is not covered here as any correlation between those levels might be estimated by monitoring/measuring, e.g. for verifying the SLA conditions.

2.4.1 Rationale for Admission Control

The following reasoning is excerpted from [ID_acaf]:

> There is a growing feeling that the basic DiffServ architectural model lacks the capability of providing service accuracy.
>
> Quoting [RFC2990]: *both the Integrated Services architecture and the Differentiated Services architecture have some critical elements in terms of their current definition which appear to be acting as deterrents to widespread deployment. There appears to be no single comprehensive service environment that possesses both service accuracy and scaling properties.* Also, in [RFC2998], it is pointed out that: *further refinement of the QoS architecture is required to integrate DiffServ network services into an end-to-end service delivery model with the associated task of resource reservation.* To this purpose, [RFC2990] recommends to *define an admission control function which can determine whether to admit a service differentiated flow along the nominated network path.* In fact, without per flow admission control, prevention of overload in a given service class, e.g. by means of pure inter-domain Service Level Agreements, does not appear to be an easy task. Upon overload in a given service class, all flows in that class suffer a potentially harsh degradation of service.

Another track of reasoning behind the introduction of admission control is that similar capabilities have been present in most telecommunication networks, in particular those based on circuit-switched principles. Hence there is a certain experience of how such mechanisms can be utilised, combining high utilisation and ensured service levels. In addition, admission control may also be seen together with the need for authorisation, implying that the means for conveying admission requests can also be applied for authorisation.

2.4.2 Overall Objectives of Admission Control

The overall objectives of having admission control are to:

- Ensure that the existing traffic flows still receive adequate service levels when additional traffic flows are introduced;
- Provide appropriate feedback/advise to a user/application when initiating a session that the session (or traffic flow) may well receive a too low service performance;
- Enable differentiation between traffic flows, including applications and users in accordance with policy and subscription/user profile;
- Balancing ensured service provision (with effective guarantees on performance levels) and efficient utilisation of network resources.

These objectives will not be equally weighed independent of the scope of discussion. For instance, from a user perspective, less interest could be placed on the network utilisation issue.

In broad terms, traffic flows can be characterised as elastic or streaming (inelastic), meaning the latter has stronger requirements on delay and delay variations. Commonly, TCP is used for the former, while UDP is used for the inelastic flows. Even though the elastic traffic flows adapt to the network conditions (to a certain extent), the effective throughput would decrease when congestion appears. Hence, packets may experience time-out, being retransmitted and the duration of the session prolonged. The ultimate factor then limiting the traffic demand is the user impatience (as it takes too long to complete the operation). It is discussed in [Robe01] that introducing some form of admission control for elastic flows may give a more effective overload control compared to relying on user impatience. A criterion for the admission decision would be to reject a new flow when the resulting bandwidth is below a specified threshold. This implies that the available bandwidth has to be followed by measurements and that the bandwidth requirement of a new flow has to be estimated/predicted.

Considering elastic traffic, the TCP flow control algorithm, applying a closed loop approach, will restrict the throughput. Naturally, this will also be influenced by the access rate and duration of the flows. An approximate expression of the effective throughput as a function of the round-trip-time and packet loss ratio has been suggested. The delay and packet loss of a flow is greatly influenced by the number of flows in progress using the same set of resources. This means that the packet scale performance of a given flow is strictly determined by flow level dynamics (i.e. number of flows and their characteristics). A simple model of this is quoted in [Robe01], referring to the processing sharing analogy when looking at a single resource. Although still being a simplified model, two central observations are made: i) the performance depends primarily on expected traffic demands and only marginally on parameters describing distribution of file lengths; and ii) performance tends to be excellent as long as expected demand is less than available capacity. The latter proposes that service differentiation can only be effectively obtained for a limited area of the workload when there are several service classes to be handled with their corresponding requirements.

For streaming traffic, the algorithms in TCP may not be in effect, e.g. as UDP could be applied. This means that the characteristics would be more determined by the inherent nature of the traffic source (as an open-loop control would be present). Thus, it is simpler for a source/application to specify the required transfer service, which for instance can be input to an admission control function (as well as resource reservation and others).

Integrating elastic and streaming traffic on the same resource units may allow for increased efficiency. By giving priority of the streaming flows, they could experience a resource that is (almost) loaded as if they were the only active flows. Then, elastic flows could be served whenever the resource is not used by the streaming flows. However, this may introduce long delays for the elastic flows during some periods. An approach is to restrict the load from the streaming flows, ensuring that some capacity is available for the elastic flows. This is an argument for introducing admission control that operates also on streaming flows.

When the capacity of the resource is limited, it is generally assumed that some form of admission control has to be present to ensure that active streaming flows receive the delay and packet loss requirements they demand. To avoid keeping a detailed list per flow, a measurement-based approach could be used, operating on the aggregated flow. It is argued in [Robe01] that such an aggregated measure might not be very precise, as the elastic traffic flows would tolerate some variation in their available service rate.

In some cases it is argued that so-called overprovisioning may make the need for traffic handling mechanisms obsolete, including admission control. Apart from the economic argument, it might also be difficult to provide the amount of capacity on certain portions, like on the access line if no technical solution is available. Another argument is the request for service differentiation. As quite a few models show, a "pure" DiffServ model may have a narrow scope for efficient differentiation that is close to an overload situation. Hence, other means for differentiation would be asked for. Therefore, providing differentiated admission criteria is one group of means that could be introduced.

2.4.3 Admission Control Taxonomy

A number of issues for describing an admission control mechanism are described in this section. The issues are not independent as some combinations may be preferred or even needed for the admission control procedure to function.

- Dynamic versus static. A dynamic mechanism is able to adapt to the changing situation, for example captured by measuring traffic loads. It is obvious that measurements for better assessing the link utilisation and characteristics of the traffic flows will lead to improved throughput. On the other hand, running continuous measurements may be fairly demanding on the routers. Hence, finding adequate measurement arrangements is a central challenge.

- Explicit versus implicit. When explicit control is used, related information is exchanged between the end system and the network. That is, protocol elements are defined which express the request for resources and the granting/rejection of resources. In case implicit control is applied, there will be no information exchange before the information transfer. An example is when the source simply starts to send packets and the network decides whether or not these packets can be forwarded without informing the source of the decision. Thus, the source has to apply other means for finding out if the transfer was successful or not (e.g. time-out on acknowledgements).

- Scope and objective function applied. When deciding whether or not to accept a request, different scopes and different sets of variables can be implemented. This is further described below.

- Traffic flow aggregates and characteristics. The admission control may work on a range of traffic flow aggregates and use various means for describing the traffic flows. Examples of bit rate measures are peak rate, mean rate and

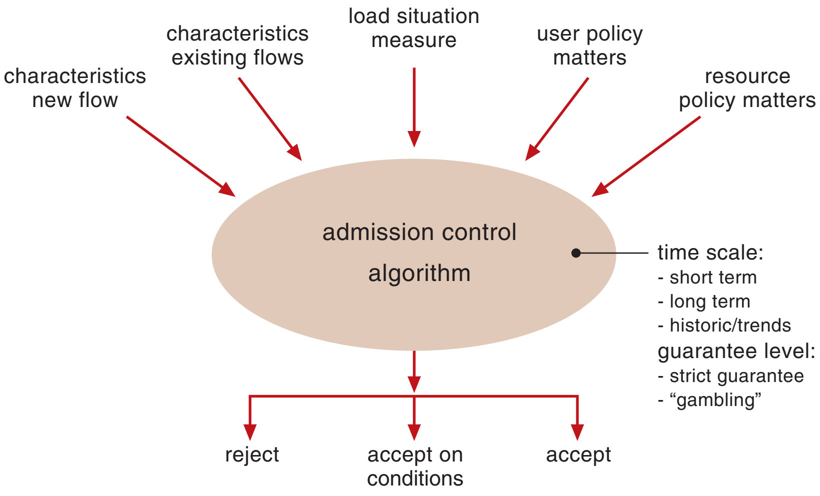


Figure 11 Input and output candidates for the admission control algorithm

equivalent rate. The latter being a measure trying to capture the variability of the rate, possibly including other aspects, like information loss ratio. Aggregates may also refer to all traffic flows related to the same session.

- Information conveyance and locating functionality. Where are the functions located and how is the information carried between the different functions? Functions could be placed in the terminals/hosts, in the edge routers, in a dedicated server, and so forth. RSVP is a protocol promoted for carrying the information between the functions, although others have also been promoted.

As mentioned earlier, executing the admission control algorithm would basically provide an answer whether to accept or reject a request for serving a traffic flow. In principle the answer could also be to accept but only on certain conditions, e.g. that some characteristics of the existing flows have to be changed/renegotiated.

Then in order for the algorithm to arrive at that decision, a number of inputs has to be available. Hence, the algorithms may differ in terms of which inputs that are needed/taken into account. Furthermore, the algorithms may also differ in terms of which answers are possible (only accept or reject, or more subtle outputs). An illustration of inputs and outputs is given in Figure 11.

As shown, some likely inputs are:

- Characteristics of the new flow (which triggers execution of the admission control algorithm). A number of parameters can be used to characterise the new traffic flow, like peak bit rate, mean bit rate, requirements on delay, jitter and loss ratio, burst size, and so forth. These could be declared by the source or, in principle, derived from other identifiers like combinations of addresses, port numbers, interfaces, etc.

- Characteristics of existing flows. The existing flows may be characterised in a manner similar to the measures used for the new flow. The measures may be declared by the sources or estimated by monitoring.

- Measure of current load pattern for the resources considered. The current load on the resource can be monitored in order to obtain a more accurate measure of the situation. Applying such an input is commonly referred to as having a measurement-based procedure.

- User policy matters. A user profile may be available, e.g. stating on what service levels and under what conditions a traffic flow for that user is to be accepted. Some criteria are time-of-day, IP address, port number, interface identity, current load situation and characteristics of the new traffic flow.

- Resource policy matters. Certain policies on how to utilise the resources may be applied, such as acceptable load levels, use of overbooking, mixture of traffic flow types, and so forth.

For the admission algorithm various scopes and principles could be valid, such as:

- What time scale is considered: Is only the current situation taken into account or is a more future-looking approach followed? Is the decision to be based also on historic/trend information? An example is that a traffic flow accompanied by low revenue could be rejected even with sufficient capacity available if there is a high probability that a flow accompanied by higher revenue will have to be rejected later on.

- What level of "gambling" is used for guaranteeing the service level? Rather loose thresholds have to be used if strict guarantees are given, while tighter thresholds (and even overbooking) could be used when a more "gambling"-like attitude is taken.

2.4.4 Implementing Admission Control

In this section, a few examples are described of how admission control can be implemented. Each of them may not be completely satisfactory for all the actors involved (e.g. user and network operator). Moreover, some of them could well be combined, like the RSVP-based and the policy-based.

RSVP-based

As a general signalling protocol, RSVP may carry most of the data needed for admission control, including characteristics of the traffic flow (see Section 7.1) as well as information about the users/port numbers.

Initiating the RSVP messages by the end systems, the traffic handling mechanisms may be co-ordinated dynamically along the relevant data path. In some places this is referred to as dynamic topology-aware admission control.

RSVP is used by an end system to request specific service levels from the network for particular traffic flows. Routers also apply RSVP to forward requests to all nodes along the path(s) of the flows and to establish and maintain state to provide the requested service. Hence, RSVP requests will generally result in resources being reserved in each node along the path. RSVP allows users to obtain preferential access to network resources, under the control of an admission control mechanism. Such admission control is often based on user or application identity; however, it is also valuable to provide the ability for per-session admission control. In order to allow for per-session admission control, it is necessary to provide a mechanism for ensuring that an RSVP request from an end system has been properly authorized before allowing the reservation of resources. In order to meet this requirement, there must be information in the RSVP message, which may be used to verify the validity of the RSVP request. An example is to have an authorization element assigned to the user, which can be inserted in the RSVP messages.

Policy and Bandwidth Broker-based

Policy and Bandwidth Broker (BB) is described in [Jens01a]. Although RSVP supports the ability to convey requests allowing for resource reservations, an essential feature may be missing. This feature is the ability of network managers and service providers to monitor, control, and enforce the use of network resources and services based on policies derived from criteria such as the identity of users and applications, traffic/bandwidth requirements, security considerations, and time of day/week. A framework for policy-based control over admission control is described in [RFC2753].

Implicit Admission Control/Local Probing

The local probing approach relies on generating test packets (probes) to check whether or not a new traffic flow can be set up. The probes may be generated by the end systems. In case several service classes are offered, it is to be decided if the probes should be sent in the same class as the following traffic flow or in another class (e.g. the lower service class). Hence, the local probing may be suitable for DiffServ. An advantage of this method is that no changes are needed in the routers not generating probes. This has also been referred to as distributed admission control, see e.g. [Kell00].

Probing results may also be based on marking (e.g. using ECN) of ongoing traffic flows. Hence, information on the marking tells the admission control algorithm whether or not a new traffic flow with certain characteristics can be served.

A common feature of implicit admission control is that no per-flow state information is needed, which may also be run in the end systems. However, remembering the connectionless nature of IP, and if the routers are not taking part in the control, it may be uncertain whether all packets in the traffic flow actually traverse the same path. This means that some mid-flow packets may well experience other conditions than the information estimated from probes if they are transferred on another path.

The different schemes for admission control may be combined. For example, an implicit admission control may be used in the access network

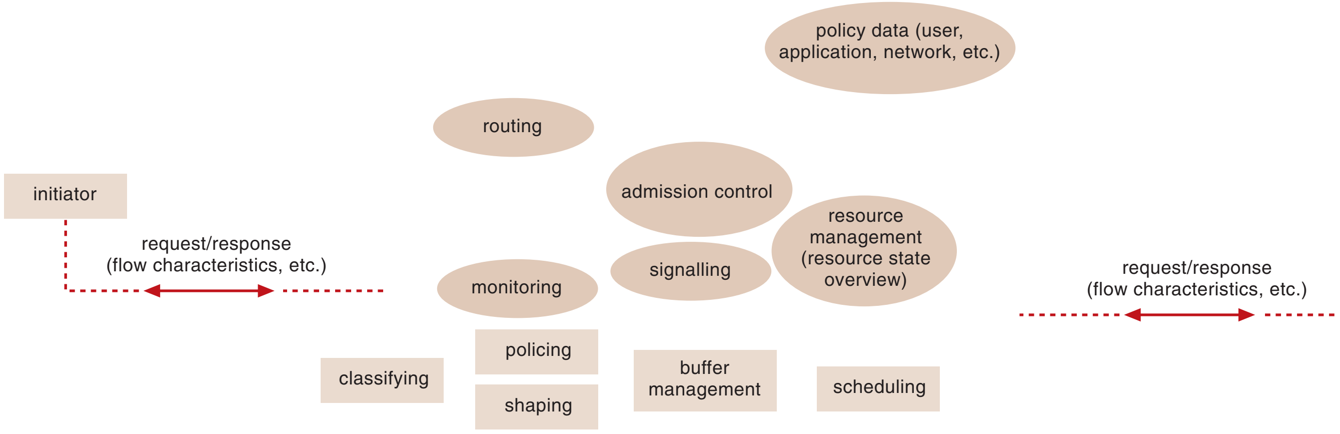
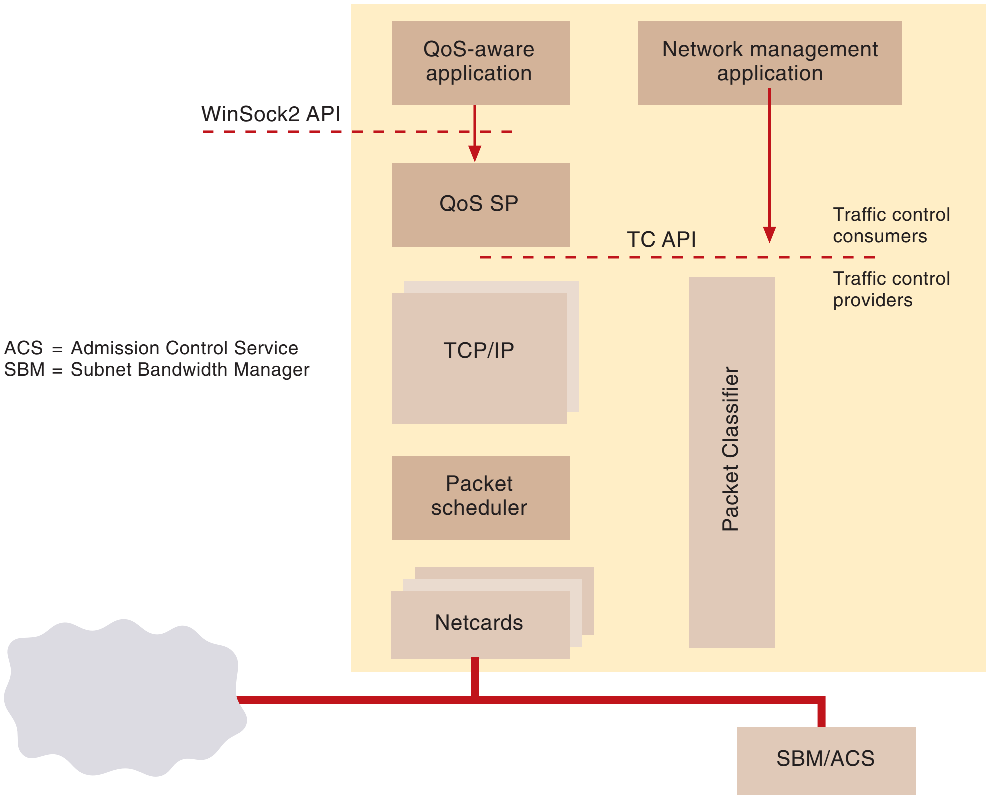

policy data (user, application, network, etc.)
routing
initiator
admission control
request/response (flow characteristics, etc.)
resource management (resource state overview)
signalling
monitoring
request/response (flow characteristics, etc.)
policing
classifying
buffer management
scheduling
shaping

Figure 12 Illustration of features related to admission control (by the network)

Figure 13 Microsoft QoS components, from [Bern00]


QoS-aware application
Network management application
WinSock2 API
QoS SP
TC API
Traffic control consumers
Traffic control providers
ACS = Admission Control Service
SBM = Subnet Bandwidth Manager
TCP/IP
Packet Classifier
Packet scheduler
Netcards
SBM/ACS

(between the terminal/host and the edge router) while other schemes are used in other parts of the network.

2.4.5 Functions Related to Network-centric Admission Control

In order to implement an explicit full-guarantee admission control, the features indicated in Figure 12 should be available. Note that some of these may be optional depending on the configuration and the overall solution for traffic handling.

Pivotal for having admission control is naturally to implement an algorithm making the admission decisions (accept/reject). In order to carry out such a ruling, the status of the resource situation (and the present load) as well as characteristics of the flow related to the request have to be made available, i.e. given as input. As described earlier this information could be provided by different means and with various levels of details and dynamics. Additional inputs may also be relevant, like user/application profiles that could be part of policy matters. Another function needed is called classification, i.e. that the packets arriving are recognised as belonging to the traffic flow in question.

In addition to the mechanisms present in the network, the terminal/application must also be able to formulate, categorise and convey the relevant information. An example of blocks adopted from Windows/Microsoft is depicted in Figure 13.

The Subnet Bandwidth Manager (SBM) can be considered as a server connected to the LAN controlling the bandwidth usage between the different hosts connected. The SBM is presented in [RFC2814]. This can be considered as a signalling protocol supporting admission control over IEEE 802-type networks by utilising RSVP. Hence, it provides a method for mapping signalling protocols, like RSVP, onto the IEEE 802-type of networks, including operations of terminals and routers in order to allow for reservation of LAN resources.

From [Bern00a] admission control agents may be allocated at key locations, referring to congestion points. Examples of such are:

- Single interface: a classic RSVP model could be applied.
- DiffServ domain: admission control at ingress router may be introduced.
- 802-based domain: Subnet Bandwidth Manager (SBM) could be introduced, ref. [RFC2814].
- ATM subnetwork: admission control at ATM edge devices
- Provider domain: admission control as part of bandwidth broker.

Still in the end system, a congestion manager may be implemented as described in [ID_cm].

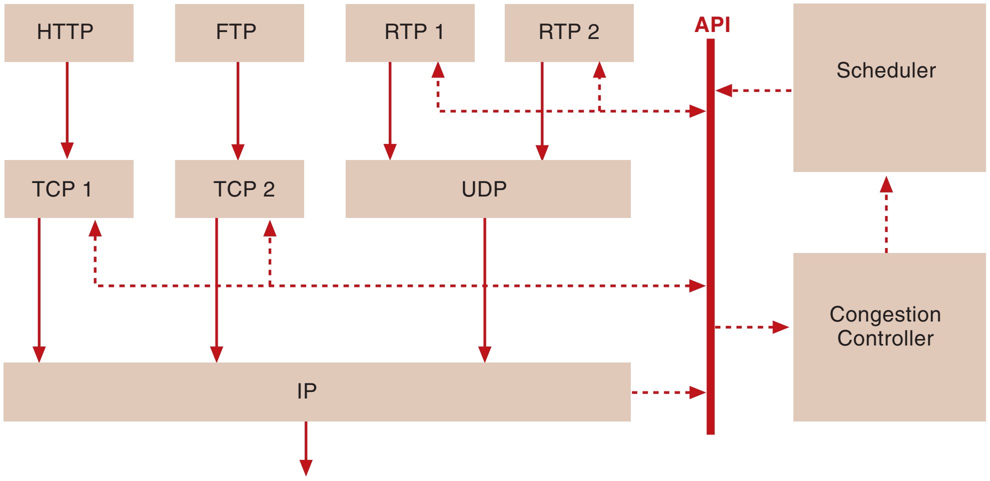


Figure 14 Framework for congestion manager, adapted from [ID_cm]

This is to support multiple traffic flows between the same sender and receiver(s) allowing the application to adapt to congestion. A framework is stretched out in that document, integrating congestion management for all types of applications and transport protocols. This is done by maintaining parameters reflecting the network condition, like throughput, round-trip delay, etc. and making this information accessible from the applications through an API as shown in Figure 14.

The main components, as depicted, are the API, the congestion controller and the scheduler. The congestion controller adjusts transmission rates based on estimates of the network condition, which is obtained from the applications (via the API). The scheduler divides the bandwidth amongst the different traffic flows

3 Best-Effort

From the outset a single class of serving IP packets was present. At the same time this was referred to as best-effort, implying that each node along the path was doing its best to transport the packet towards its destination. A fairly simple router implementation may then be adequate as schematically depicted in Figure 15.

Packets entering the router are to be forwarded to the output link. Buffers (queues) may be implemented at several places. Basically a single queue may suffice for each link, being served according to a first-in-first-out discipline.

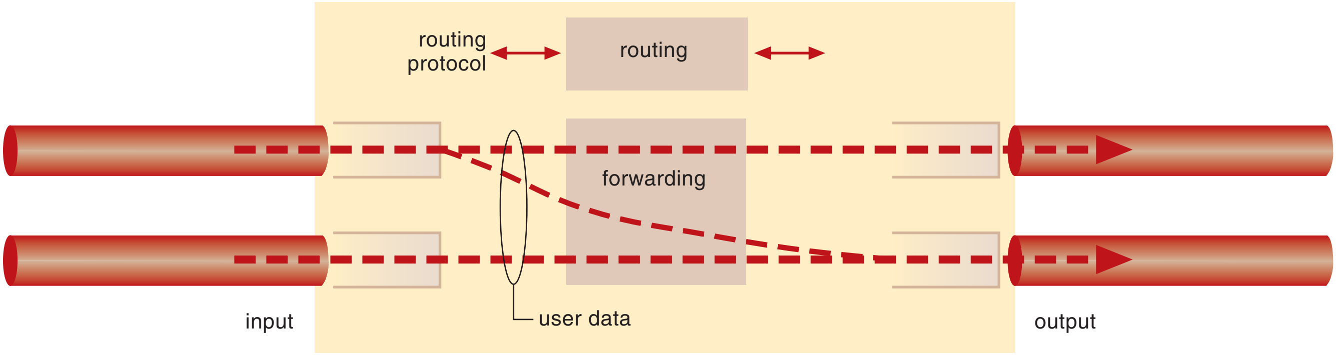
As also shown, a separation between *forwarding* and *routing* is made; routing referring to exchanges of routing information (by routing protocols) to set up routing tables, and forwarding referring to sending packets on according to the information in the routing tables.

The traffic flows carried by the node may have different characteristics, e.g. in terms of packet lengths, bit rates, use of transport protocols, and so forth. Combining all these in the same buffer and on the same link may pose additional challenges, as described in several accompanying papers in this issue of *Telektronikk*. Hence, other service models – Differentiated Services and Integrated Services – have been defined, as treated in the following chapters.

4 Differentiated Services

Differentiated Services (DiffServ) is promoted as a service architecture supporting a scalable way to achieve relative service and QoS levels in an IP network. DiffServ operates on aggregated flows by dividing the traffic flows into a set of classes. The DiffServ architecture is defined in [RFC2475], see Figure 16.

Figure 15 Schematic illustration of router for best effort



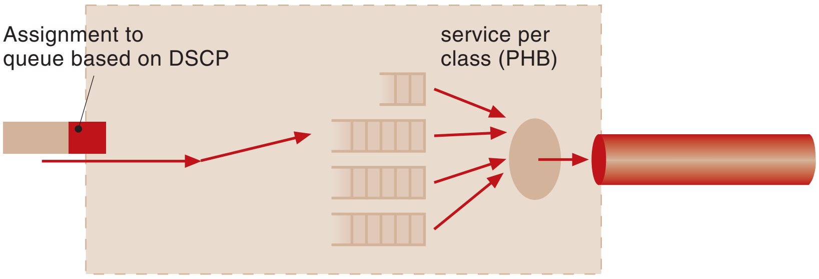


Figure 16 DiffServ uses service classes (traffic aggregates)

As explained in [Jens01] DiffServ uses a particular implementation of the IP version 4 Type of Service (ToS) header field. This field is now called the DiffServ field, consisting of 8 bits, out of which 6 bits are available for current use and two are reserved for future use. The 6 bits define the *DiffServ Code Point* (DSCP), which identifies a *Per-Hop Behaviour* (PHB). The PHB indicates the way packets shall be handled in the routers and can be set and reset in any DiffServ capable node, also referred to as marking the IP packet. Some standardised PHBs are: default class (DE, [RFC2474]), Class Selector (CS, [RFC2474]), Expedited Forwarding (EF, [RFC2598]), and Assured Forwarding (AF, [RFC2597]).

EF is described as a forwarding treatment for supporting a low loss, delay and jitter end-to-end service with assured bandwidth. The exact way to implement such a service by mechanisms in the network is left open for the operators/vendors.

AF is a group of PHBs. The idea is to support services with requirements for assured packet delivery. It is assumed that customers have subscription-based traffic profiles, Service Level Specification (SLS). Packets within the profile shall have a high probability of delivery, whereas out-of-profile packets can be delivered if bandwidth is available in the network. For this purpose packets are given different drop precedences. The AF group defines four classes of traffic. For each class three drop precedences are defined, each representing a PHB and thus a reserved DSCP value.

A central term in DiffServ is a *Behaviour Aggregate* (BA). This is the aggregation of all packets with the same DSCP and crossing a given link in a particular direction [RFC2475]. The set of BAs sharing an ordering constraint is called an *Ordered Aggregate* (OA). For example, all packets belonging to a given AF class and crossing a given link in a particular direction share an ordering constraint. This is because the AF definition states that AF packets of the same microflow belonging to the same AF class must not be reordered (their sequence must not be changed) regardless of their drop precedence.

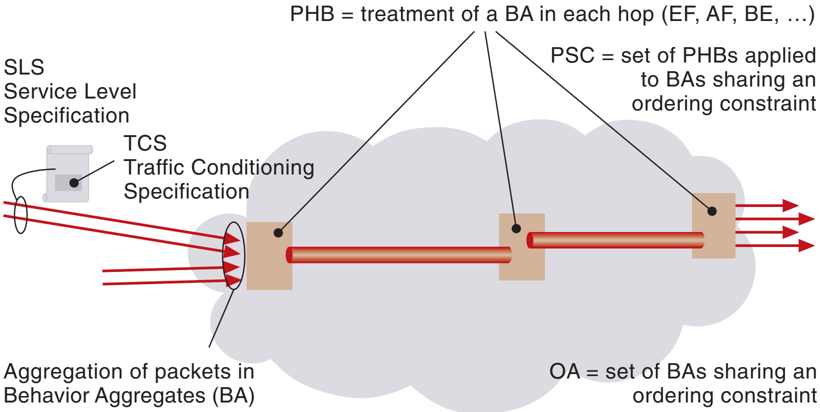
Another term is the *Per-Hop Behaviour Scheduling Class* (PSC). A PSC is the set of PHBs that are applied to the BAs belonging to an OA. For example, the PHB that is associated with a given AF class constitute a PSC. Hence, PSC is a PHB group for which a common constraint is that ordering of at least those packets belonging to the same microflow must be preserved.

A *Service Level Specification* (SLS) is defined ([ID_dsterms]) as a set of parameters and their values which together define the service offered to a traffic stream by a DS domain. An integral element of an SLS is the *Traffic Conditioning Specification* (TCS). The TCS is defined as a set of parameters and their values which together specify a set of classifier rules and a traffic profile. These terms are illustrated in Figure 17.

The terms in the SLS are checked at the border of the domain, e.g. in an edge router. In case the appropriate DSCP value has not been inserted in the packet, this has to be done, based on various combinations of information in the packet. This information may include the IP packet header as well as the header of the transport protocol. Additional information may also be used, like the interface on which the packet arrived on. This means that a *Multi-Field* (MF) classifier and marker would be activated in the first DiffServ-capable router in the domain. Then the packet has got its DSCP implying that a BA is given. Characteristics of a flow (aggregate) will then be monitored to see whether the packet is forwarded directly (conditions in the SLS are obeyed), being dropped, re-marked or shaped. A logical relation between the different functions is illustrated in Figure 18.

A distinction is made between an MF classifier and a BA classifier. The MF classifier classifies packets based on one ore more fields in the packet. This is normally only done at the edge of the network. The BA classifier classifies packets

Figure 17 Some terms related to DiffServ



based solely on the DSCP value. This can take place in every DiffServ-capable router.

Which functions that are activated in a router would depend on where the router is located, e.g. ingress, egress or interior. An example of use of functions is given in Table 1.

One is faced with quite a few challenges when deciding to mark packets:

- Applications may use transient ports or source multiple traffic flows on the same port, where the flows may require different service levels.
- Users' IP addresses change as a result of DHCP. Multi-user machines use the same IP address for multiple users.
- IPSec encryption encrypts port identities, leaving them less useful as classification criteria.

Basically, the end system may mark the packets, or the edge router may do the marking. Allowing end systems to mark the packets would likely make it easier to meet the challenges listed above.

5 IntServ

The Integrated Services (IntServ) architecture was defined to allow separate treatment to individual or groups of traffic flows, [RFC1633]. Two sets of capabilities are necessary to enable this: i) functions in individual network elements along the path; and ii) ways to communicate the requests between the network elements.

In [RFC1633] a flow is defined as a *distinguishable stream of related datagrams that results from a single user activity and requires the same QoS*. That is, it is the finest granularity of packet stream that can be identified. A flow is unidirectional (from a single source to a set of destinations). In order to identify a flow, an MF filter can be applied, as described in the previous chapter.

Box A Some DiffServ Terms

Flow – A stream of packets with the same source IP address, source port number, destination IP address, destination port number, and protocol identity (packets not separated by a time longer than a threshold).

Service Level Agreement (SLA) – A service contract between a customer and a service provider that specifies the forwarding service a customer should receive. A customer may be a user organisation or another provider domain (upstream domain).

Traffic profile – A description of the properties of a traffic flow such as rate and burst size.

Precedence Field – The three leftmost bits in the TOS octet of an IPv4 header. Note that in DiffServ, these bits may or may not be used to denote the precedence of the IP packet.

TOS field – Bits 3–6 in the TOS octet of the IPv4 header.

Differentiated Services field (DS field) – The TOS octet of an IPv4 header, or the traffic class octet of an IPv6 header is renamed the differentiated services field by DiffServ. It is the field where service classes are encoded.

Admission Control – The decision process of whether to accept a request for resources (link bandwidth plus buffer space).

Classification – The process of sorting packets based on the content of packet headers according to defined rules.

Behaviour Aggregate (BA) classification – The process of sorting packets based only on the content of the DS field.

Multi-Field (MF) classification – The process of classifying packets based on the content of multiple fields such as source address, destination address, TOS octet, protocol identity, source port number, and destination port number.

Marking – The process of setting the DS fields of packets.

Policing – The process of handling out-of-profile traffic, e.g. discarding excess packets.

Shaping – The process of delaying packets within the traffic flow to make it conform to some defined traffic profile.

Scheduling – The process of deciding which packet to send first in a system of multiple queues.

Queue management – Controlling the length of packet queues by dropping packets when necessary or appropriate.

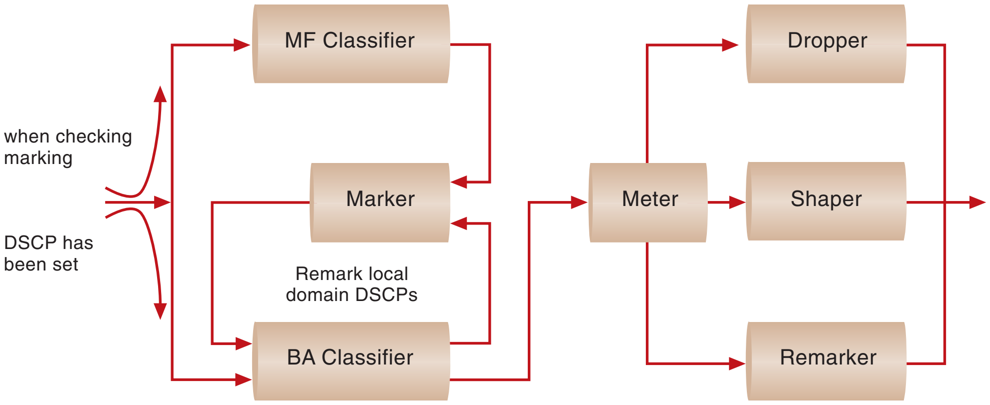


Figure 18 Logical view of traffic conditioning in a DiffServ node

Function block	Ingress router	Interior router	Egress router
MF classification	X		
BA classification	X	X	X
Meter	X		X
Marker	X		X
Policer/Dropper	X		
Shaper	X		X
Signalling	X		X

Table 1 Example of use of the function blocks related to DiffServ

Two service classes are described in addition to the best effort class; Controlled Load [RFC2211] and Guaranteed Service [RFC2212]. An objective of the Controlled Load (CL) class is to offer low average delay and limited loss. It is intended to offer roughly the same service (e.g. throughput, delay, loss) independent of the network load. Thus, if a flow is accepted for the CL service, the routers are supposed to make a commitment to offer a flow service level equivalent to that seen by a best-effort flow on an unloaded network. The CL class supports applications that can tolerate a small amount of delay and loss (e.g. adaptive real-time services). The Guaranteed Service (GS) class offers a quantifiable bounded queuing delay and no loss and is intended for real-time applications with stringent timing requirements.

In order to provide the requested service class to an application, information on class (and corresponding parameters) has to be conveyed to a network element (e.g. a router). Signalling (e.g. Resource Reservation Protocol, RSVP) can be used to create and maintain the required flow-specific states in network elements allowing them to provide the requested services. Introducing such control activities allows for additional mechanisms related to resource handling. These are depicted in Figure 19.

Prior to transferring “user” data a signalling sequence has to be carried out (optionally, this could also be done by management operations or as a combination). Upon arrival of a request (e.g. RSVP_PATH message) a router would check its current load/configuration to decide whether or not the request can be served (done by the admission control). In case the request can be served, a corresponding message is passed on to the next element. When the request has made the whole path (between the two end points), an “acknowledge” message (e.g. RSVP_RESV message) is returned if all required resources are available. When such a message arrives at a network element, the resources are reserved implying that the units depicted as part of the IP Forwarding module in Figure 19 are properly configured. Information in the admission control could also be updated. Then “user” data can be transferred (having the specified Multi-Field class) and being served as notified by the signalling activities.

IntServ nodes support the required performance guarantees by implementing multiple queues per output port in combination with scheduling and buffer management. Typical mechanisms are Weighted Fair Queuing (WFQ) scheduling and a Random Early Detection (RED) variant for congestion avoidance as described in Chapter 2.

A key feature of the IntServ model is that resources are explicitly managed; by resource reservation and admission control. According to [RFC1633] traffic control is implemented by the packet scheduler, the classifier, admission control and the reservation setup protocol:

- *Packet scheduler* manages forwarding of flows by use of buffering and scheduling algorithms. Policing is taken care of by the scheduler.

- *Classifier* maps each incoming packet to a class (all packets in the same class get the same treatment from the scheduler). A class is an abstraction that may be local to a router;

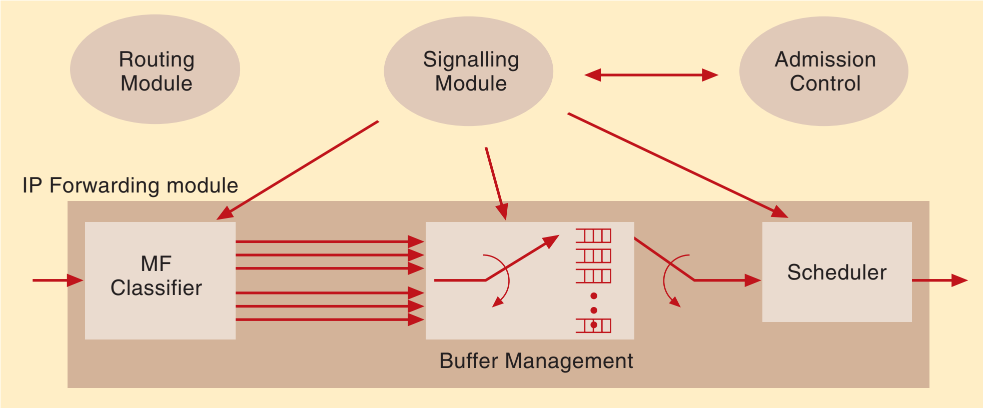


Figure 19 Schematic illustration of an IntServ router, adapted from [RFC1633]

the same packet may be classified differently by different routers along the path.

- *Admission control* implements the decision algorithm that is used to decide whether or not a new flow can be accepted without violating performance guarantees to it or existing flows.

- *Reservation setup protocol* is used to convey information between the network elements along the path. In order for an application to state its requirements, the protocol has to be able to carry the corresponding information elements (parameters).

For reservation of resources, the messages used in the setup protocol carry some fields describing the traffic flows. These are described in Chapter 7.

As mentioned, the Guaranteed Service ensures bounds on delay and throughput. In order to configure the resources along the path accordingly, the parameters conveyed by the setup protocols have to be properly quantified. Commonly, Resource reSerVation Protocol (RSVP) is used as an example of such a setup protocol. The relevant parameters are then:

- r; rate of leaky bucket (measured in bytes per second: 1 byte/sec – 40 Tbyte/sec);

- b; depth of leaky bucket (measured in bytes: 1 byte – 250 Gbyte);

- M; maximum packet size;

- m; minimum policed unit (packets shorter than m are counted as having size m);

- p; peak rate (measured in bytes per second – same as for r);

- R; service (link) rate (measured in bytes per second – same as for r);

- S; slack term (measured in µs).

The two latter are part of the *RSpec* field while the 5 first parameters are carried in the *TSpec* field, see Figure 20.

An estimate for the end-to-end delay bound is then given as, ref. [RFC2212]:

$$\text{e} - \text{t} - \text{e delay bound} =$$

$$\begin{cases} \dfrac{\frac{b-M}{R}\cdot(p-R)}{p-r} + \dfrac{M+Ctot}{R} + Dtot & \text{, if } p \geq R \geq r \\ \dfrac{M+Ctot}{R} + Dtot & \text{, if } r \leq p \leq R \end{cases}$$

Ctot and *Dtot* are “correction” terms influenced by the way the packets are treated along the path.

C refers to a rate-dependent term, i.e. it represents the delay a packet may experience because of variations in the service rate, which depend on the rate itself. An example is time for fragmenting packets into cells (e.g. ATM cells), which depends on the rate of sending ATM cells. The term *C* is measured in units of bytes.

D is a rate-independent term given per node. It captures the variation in transition time through the node (worst case). The term *D* is measured in units of µs.

Ctot and *Dtot* are found by adding contributions to the terms *C* and *D* along the path.

The basis for the estimate of end-to-end delay bound is a fluid flow model. Then, the terms *C* and *D* may be considered as capturing deviates of the node compared to a fluid flow model.

Note that the Guaranteed Service does not control minimal or average delay, nor the propagation delay, only the maximal queueing delay. Neither does the Guaranteed Service give an estimate of the jitter as such.

When aggregating and merging flows, a way of handling the traffic parameters is asked for, ref. [RFC2212]:

- TSpec for a merged flow may be calculated by: i) taking the largest token bucket rate; ii) the largest bucket size; iii) the largest peak rate; iv) the smallest minimum policed unit; and v) the smallest maximum packet size, across all flows in the merged flow. A merged TSpec is one that is adequate to describe the traffic from any one of the constituent TSpecs.

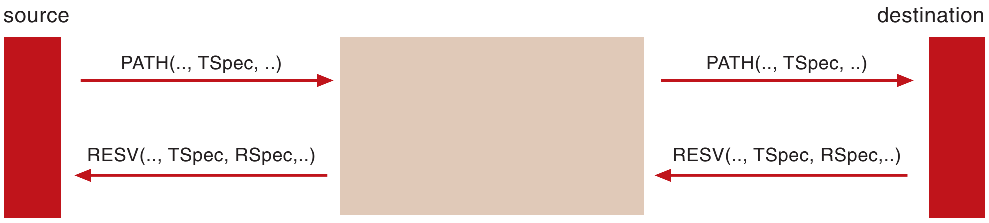


Figure 20 Messages for RSVP

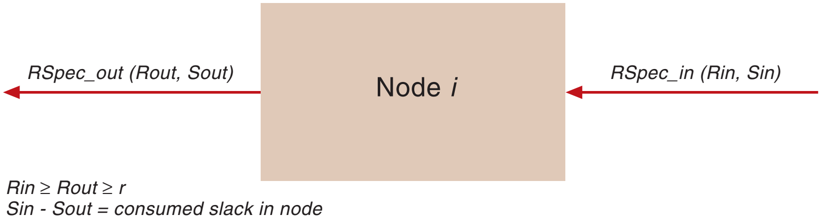


Figure 21 Relations between parameters in incoming and outgoing side of a node have to be found

- TSpec summed may be calculated by: i) the sum of token bucket rates; ii) the sum of bucket sizes; iii) the sum of peak rates; iv) the smallest minimum policed unit; and v) the maximum packet size, across all flows in the set.

- TSpec as a least common "measure" is one that is sufficient to describe the traffic of anyone in the set. It may be calculated by: i) the largest token bucket rate; ii) the largest bucket size; iii) the largest peak rate; iv) the smallest minimum policed unit; and v) the largest maximum packet size, across all flows in the set. It differs from the merged flow by the way the maximum packet size is found. Note that merge refers to characteristics of packet flows while least common refers to requests on the resource capabilities.

Values of the RSpec field can be considered in a similar way as for the TSpec field; i.e. a set of RSpecs is merged into a single RSpec by taking the largest rate R, $Rout = \max_j\{Rj\}$ and the smallest slack S, $Sout = \min_j\{Sj\}$.

Consider a node along the path, see Figure 21. When it receives a setup message containing the TSpec and RSpec fields, it has to estimate the values to assign to the RSpec parameters on the outgoing side (when the setup message is passed on to the following node).

The overall rule for determining the new RSpec values is given by the delay constraint:

$$Sout + \frac{b}{Rout} + \frac{Ctot\_i}{Rout} \le Sin + \frac{b}{Rin} + \frac{Ctot\_i}{Rin}$$

where $Ctot\_i$ is the cumulative sum of "deviation" terms, C, for all network elements in the upstream (between the destination and node i), including node i.

To ensure no loss of flow, some portion of a buffer has to be allocated. If a fluid flow model would be adequate, this amount would simply be equal to b, the token bucket size. However, taking into account that the traffic flow may gain burstiness in the network, some margin should be considered. An estimate of the needed buffer space is, ref. [RFC2212]:

buffer space =

$$M + \frac{(b-M)\cdot(p-X)}{p-r} + X \cdot \left(\frac{Csum}{R} + Dsum\right)$$

where

$$X = \begin{cases} r & \text{, if } \dfrac{b-M}{p-r} < \dfrac{Csum}{R} + Dsum \\ R & \text{, if } \left\{\dfrac{b-M}{p-r} \ge \dfrac{Csum}{R} + Dsum\right\} \wedge \{p > r\} \\ p & \text{, otherwise} \end{cases}$$

Csum and *Dsum* are considered for the aggregate using that buffer space.

Using these estimates, both expressions for assigning resources in the node and expressions for finding the parameter values in the setup message on the outgoing side are given.

In order to provide the performance guarantees as stated by the IntServ node, several queues are used per output interface, in combination with scheduling and buffer management as depicted in Figure 19. As guarantees are provided on a per flow basis for the Guaranteed Service, a separate queue per flow could be needed. However, a common queue can be used for a group of flows in case these have similar requirements on delay and loss. In particular, a common queue can be used for flows requesting Controlled Load service.

Three groups of queues may therefore be seen; one set for flows in the Guaranteed Service class, one set for flows in the Controlled Load class, and one set for flows in best effort class. The two latter classes might consist of a common queue. Appropriate scheduling mechanisms are then applied to serve the queues such that the service level guarantees given to the flows are kept.

6 Multi-Protocol Label Switching, MPLS

At the IP layer (layer 3) a router makes forwarding decisions for a packet based on information in the IP header. The analysis of the packet header is performed and an algorithm is executed in each router to decide upon further treatment. This can be viewed as a two step process, ref [RFC3031]: i) The packets are classified into a set of *Forwarding Equivalence Classes (FECs)*; ii) Each FEC is mapped to a next hop.

The following advantages of MPLS are listed in [RFC3031]:

- MPLS forwarding can be done by nodes incapable of analysing the IP packet headers, or incapable of analysing these headers with sufficient high speed.

- Assigning a packet to an FEC, the ingress router may use information about the packet that goes beyond the content of the packet header, like the interface. Hence, assignment to FECs can be a more involved process, without impacting all routers in the network.

- Forwarding decisions within a network may be made depending on which ingress router a packet used. Then a packet may be forced to follow a particular route explicitly chosen, circumventing the ordinary routing.

Some central terms for MPLS are given in Box B.

6.1 MPLS Formats and Terms

To some extent, the use of *Label Switched Paths* (LSPs) can be considered as introducing tunnelling as seen from the IP layer. That is, when an LSP is introduced an intermediate node would not examine the IP header information in order to decide upon the proper handling of the packets arriving in that LSP. That is, with MPLS the classification of packets into FECs is only performed at the ingress to the MPLS domain. The packet is then mapped to an LSP by encapsulation of an MPLS header. The LSP is identified locally by the header, see Figure 22. Based on the value of the label the packet is mapped to the next hop. In successive routers within the MPLS domain the label is swapped (therefore it can have only local significance) and the packet is mapped to the next hop.

An LSP can be considered as a path created by concatenation of one or more hops, allowing a packet to be forwarded by swapping labels from an incoming to an outgoing side of the MPLS node. An MPLS path is frequently referred to as layer 2 1/2 in the OSI model. That is, it may be considered as a tunnel as mentioned above. In order to introduce a tunnel, a "header" is attached to the IP packet as shown in Figure 22 for the Point-to-Point Protocol (PPP) case (e.g. IP over SDH). When the IP packets are carried by ATM the label may be identical to the VPI/VCI fields in the ATM cell header. The MPLS architecture is described in [RFC3031].

Box B Selected MPLS terminology (from [RFC3031])

Label – A short fixed length physically continuous identifier used to identify an FEC, of local significance.

Label merging – The replacement of multiple incoming labels for a particular FEC with a single outgoing label.

Label swap – The basic forwarding operation consisting of looking up an incoming label to determine the outgoing label, encapsulating, port, and other data handling information.

Label swapping – A forwarding paradigm allowing streamlined forwarding of data by using labels to identify classes of data packets which are treated indistinguishably when forwarding.

Label switched hop – The hop between two MPLS nodes, on which forwarding is done by use of labels.

Label switched path – The path through one of more Label Switched Routers (LSRs) at one level of the hierarchy followed by a packet of a particular FEC.

Label stack – An ordered set of labels.

Merge point – A node at which label merging is done.

MPLS domain – A continuous set of nodes which operate MPLS routing and forwarding and which are also in one routing or administrative domain.

MPLS edge node – An MPLS node that connects an MPLS domain with a node which is outside of the domain, either because it does not run MPLS, and/or because it is in a different domain. Note that if an LSR has a neighbouring host which is not running MPLS, that LSR is an MPLS edge node.

MPLS egress node – An MPLS edge node in its role in handling traffic as it leaves an MPLS domain.

MPLS ingress node – An MPLS edge node in its role in handling traffic as it enters an MPLS domain.

MPLS label – A label which is carried in a packet header, and which represent the packet's FEC.

MPLS node – A node which runs MPLS. An MPLS node will be aware of MPLS control protocols, will operate one or more layer 3 routing protocols, and will be capable of forwarding packets based on labels.

Referring to Figure 22, the fields in the MPLS header can be used as follows:

- Label – contains a 20 bit tag identifying an LSP;

- Exp – contains 3 bits (originally not allocated, intended for experimentation) which can refer to a certain service class, e.g. in analogy to the DiffServ classes;

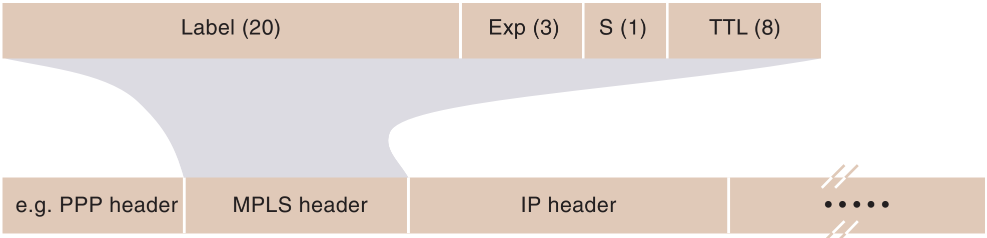


Figure 22 MPLS header and placement in layer "2 1/2"

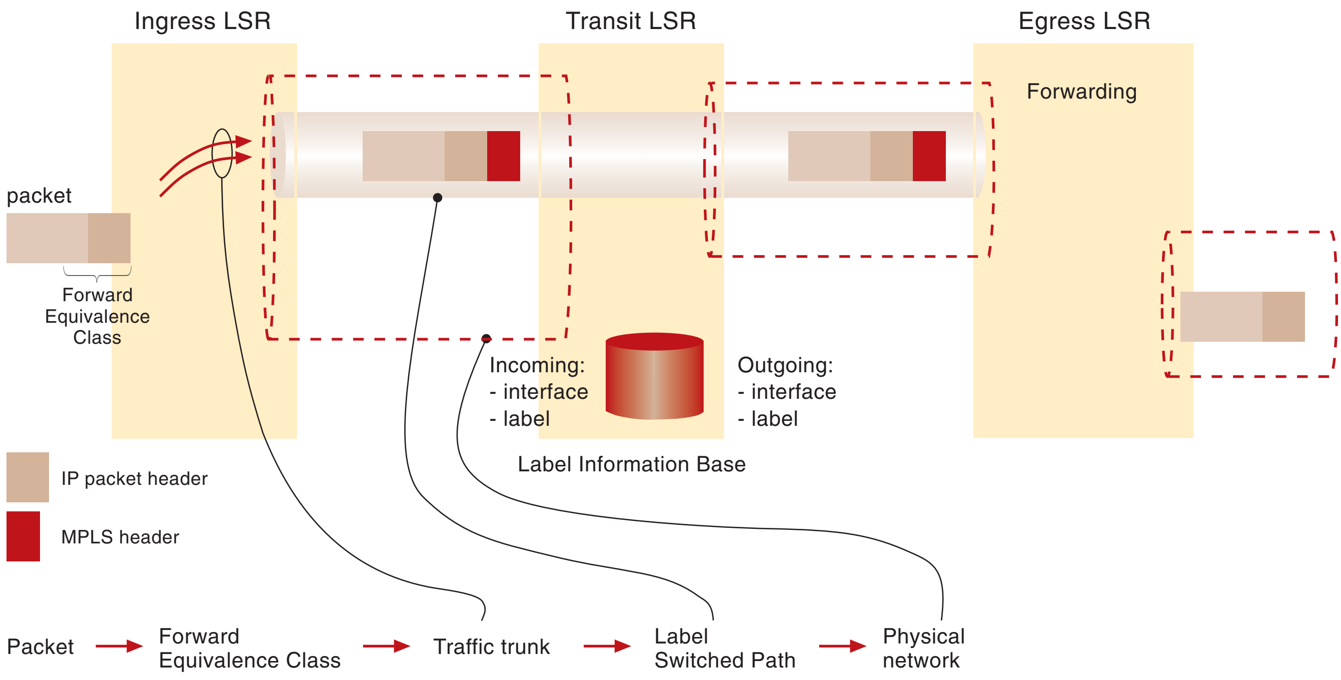


Figure 23 Illustration of information attached to an LSP in an LSR

- S – 1 bit indicates end of label stacking as several labels may be stacked;

- TTL – 8 bit giving the Time To Live information.

When an MPLS packet enters a *Label Switching Router* (LSR) a table containing information, *Label Information Base* (LIB) on further treatment of the packet is looked up. This is illustrated in Figure 23. This base may also be referred to as the *Next Hop Label Forwarding Entry* (NHLFE), which typically contains the following information (ref. [RFC3031]):

- Next hop of the packet;

- Operation to perform on the packet's label stack (replace the label at the top with another label, pop the label stack, or replace the label at the top of the stack with a new label and at the same time push one or more new labels onto the stack);

- Data link encapsulation to use when transmitting the packet;

- Way of encoding the label stack when transmitting the packet;

- Other information relevant to forwarding treatment.

In a given LSR, the "next hop LSR" may be the same LSR, implying that the top level label should be popped and the packet "forwarded" to itself, allowing or more forwarding decisions.

At the ingress of an MPLS domain an FEC-to-NHLFE mapping is needed, that is when packets arrive without an MPLS label.

Within an MPLS domain an incoming label mapping is executed, mapping the packet onto a set of NHLFEs.

MPLS can operate on a label stack. Operations on this stack are push, pop and swap. This can be used to merge and split traffic streams. The push operation adds a new label at the top of the stack and the pop operation removes one label from the stack. The MPLS stack functionality can be used to aggregate traffic trunks. A common label is added to the stack of labels. The result is an aggregated trunk. When this MPLS path is terminated the result will be a splitting (de-aggregation) of the aggregated trunk into its individual components. Two trunks can be aggregated in this way if they share a portion of their path. Hence, MPLS can provide hierarchical forwarding, which may become an important feature. A consequence may be that the transit provider need not carry global routing information, thus making the MPLS network more stable and scalable than a full-blown routed network.

To limit the number of MPLS paths, merging can be utilised. Then two paths in the same direction and with common requirements are placed together in a common LSP on the outgoing side resulting in a many-to-one mapping of labels.

6.2 TE and MPLS

An explicitly routed LSP is an LSP whose path is established by means other than normal IP routing. In one approach this requires among other things a management system representation as described in [Henr01].

When utilising MPLS with Traffic Engineering, a number of mapping relations is asked for, see Figure 23:

- Mapping packets onto FECs. An FEC composes a group of packets to be forwarded over the same path with the same forwarding treatment. In order to carry out this mapping fields in the IP packet are examined.

- Mapping FECs onto traffic trunks. A traffic trunk is an aggregation of traffic flows of the same class. A traffic trunk can again be routed (placed inside an LSP, i.e. a traffic trunk is only given for one LSP and not a sequence of LSPs).

- Mapping traffic trunks onto LSPs.

- Mapping LSPs onto links in the physical network.

In several sources, the terms traffic trunk and LSP are used synonymously. However, a fundamental difference between traffic trunk and LSP can be observed; that is, a traffic trunk is an abstract representation of traffic to which specific characteristics can be associated. An LSP is a description of a path in the network through which the traffic traverses.

Trunks having the same egress point may be merged into a common tree. This may reduce the number of trees significantly. Trunks can also be aggregated by adding a new label to the stack for each trunk (that is, bundling the trunks into a single path/tunnel).

Designing an MPLS network "on top of" a physical network could be looked upon as relating two graphs to each other;

- Physical graph, $G = (V, E, c)$, is a capacitated graph depicting the physical topology of the network. V is the set of nodes in the network and E is the set of links. For v to w in V, (v, w) represents the link in E when v and w are directly connected under G. c indicates the set of capacity and other constraints associated with E and V.

- MPLS graph, $H = (U, F, d)$, where U is a subset of V representing the LSRs, that is the set of LSRs that are end point of at least one LSP. F is the set of LSPs. For x and y in U, (x, y) is in F if there is an LSP going from x to y. d represents the set of demands and restrictions associated with F.

The fundamental problem of designing an MPLS network is to relate the two graphs such that an objective function is optimised. This is addressed in several accompanying papers in this *Telektronikk* issue.

One of the requirements from Traffic Engineering is to be able to reroute an LSP under a number of conditions (failure, better route available, etc.). This should preferably be done without disturbing the traffic flows; for example by establishing the new LSP before the old/existing LSP is released, which is called make-before-break. In case the existing and new LSP compete for the same resources, particular concerns have to be made, also considered by the admission control.

As mentioned above, IP packets are classified at the ingress of the MPLS domain into a number of Forwarding Equivalence Classes (FECs). All the packets in a given FEC are treated the same way within the domain. Choosing the use of FEC may depend on criteria like:

- the user (derived from the source address, interface, etc.);
- the application type;
- the packet destination.

A traffic trunk is described by its ingress and egress LSRs, the set of FECs which is mapped onto it, and a set of attributes that gives its characteristics. Two fundamental questions have to be answered related to traffic trunks; i) how to give the characteristics; and ii) how to relate traffic trunks to the physical network (through LSPs). This requires three capabilities:

- Set of attributes characterising the traffic trunks that give its characteristics;

- Set of attributes related with resources that constrain the placement of traffic trunks onto the resources;

- Mechanism for placing/maintaining traffic trunks onto the set of resources.

For the last item, constraint-based routing could be applied as described in [Feng01].

Attributes characterising traffic trunks are ([RFC2702]):

- Traffic parameter attributes. These are used to describe the traffic flows (the FECs) transported in the traffic trunk. Relevant parameters include peak rates, average rates, maximal burst size, etc. Possibly, equivalent measures could be applied, like the effective bandwidth.

- Explicit path specification attribute. An explicit path assignment for a traffic trunk is a path that is specified through "operator" action (e.g. management procedures). Such a path can be completely or partially specified. Path preference rules may be associated with explicit paths, telling whether the explicit path is "mandatory" (has to be followed) or "optional" (other paths could be selected in case sufficient resources are not available on the preferred path).

- Resource class affinity attribute. This attribute can be used to specify which resource types that can be explicitly included or excluded from the path through which the traffic trunk is routed. If no affinity attribute is given a "don't care" condition is assumed. Routing traffic trunks onto resources have to take these attributes into account, matching the requirements.

- Adaptivity attribute. As network state and traffic state change over time, more optimal routes of traffic trunks could appear. Setting this attribute tells whether or not the route can be re-optimised for the traffic trunk. However, appropriate thresholds should be given avoiding too frequent changes of routing.

- Load distribution attribute. In case several traffic trunks are used between the pair of nodes, the load distribution attribute can tell whether or not the load (traffic trunk) can be distributed on these trunks. In general the packet order should be maintained, implying that packets belonging to the same traffic flow are transferred on the same traffic trunk.

- Priority attribute. This attribute gives the relative importance of the traffic trunk. The value can be used to determine the order in which trunks are assigned to paths under establishment and failure situations. Priorities will also be used together with pre-emption.

- Pre-emption attribute. The value of this attribute tells whether or not a traffic trunk can preempt another traffic trunk, and whether or not another traffic trunk can preempt a specific traffic trunk. This will assist to ensure that high priority traffic trunks are routed through even though the capacity is not sufficient to handle all traffic trunks.

- Resilience attribute. The resilience attribute gives the behaviour of a traffic trunk when faults occur along the path followed by a traffic trunk. In case of fault the traffic trunk could be rerouted or not depending on the value of this attribute. For rerouting, the constraints given (e.g. by affinity) could be observed or not.

- Policing attribute. The value of this attribute tells which actions to take when the traffic on the trunk is non-compliant (excessive traffic). Examples of actions are packet dropping (rate limiting), packet tagging and packet shaping.

In addition to attributes related to traffic trunks, some attributes are also related to resources (frequently thought of as the links). These attributes are ([RFC2702]):

- Maximum allocation multiplier attribute. The value of this attribute tells what proportion of the link and buffer capacity that is available. Then "over-allocation" could be achieved (as well as "under-allocation").

- Resource class attributes. The attributes express the resource type (e.g. thought of as colours). These are matched with the traffic trunk affinity attribute when finding paths onto which the traffic trunks are routed.

Basic operations on traffic trunks are ([RFC2702]):

- Establish a traffic trunk;
- Activate a traffic trunk, to start forwarding packets;
- Deactivate a traffic trunk;
- Modify attributes for a traffic trunk;
- Reroute a traffic trunk;
- Remove a traffic trunk.

In addition to these basic operations, a few more, like policing and shaping could also be defined.

A traffic trunk is defined as unidirectional. As a bidirectional transfer capability is commonly asked for, two traffic trunks having the same end points but passing packets in opposite directions can be defined. In case these are always handled as a unit, it is called a bidirectional traffic trunk (they are established as an atomic operation and one may not exist without the other). If a trunk is routed through a different physical path than the corresponding trunk in the opposite direction, the bidirectional traffic trunk is called topological asymmetric. Otherwise, it is called topological symmetric.

MPLS is an essential component for carrying out Traffic Engineering in IP-based networks. A simple argument is its inherent feature to circumvent the "ordinary" IP packet handling in traversed routers. According to [RFC2702], the advantages of MPLS can be further described by:

- Explicit label switched paths which are not constrained by the destination based forwarding paradigm can easily be created through administrative action or through automated actions by the underlying protocols.

- LSPs can potentially be efficiently maintained.

- Traffic trunks can be instantiated and mapped onto LSPs.

- The attributes can be associated with traffic trunks that modulate their behavioural characteristics.

- A set of attributes can be associated with resources that constrain the placement of LSPs and traffic trunks across them.

- MPLS allows for both traffic aggregation and disaggregation whereas classical destination only based IP forwarding permits only aggregation.

- It is relatively easy to carry out constraint-based routing with MPLS.

- A good implementation of MPLS can offer lower overhead than competing alternatives for Traffic Engineering.

One way of designing MPLS networks is to apply constraint-based routing. Then the following information is commonly used as input:

- Attributes associated with traffic trunks;
- Attributes associated with resources;
- Other topology state information.

Based on this every node may find an explicit route for each traffic trunk originating from that node. Here, an explicit route for each traffic trunk is a specification of an LSP that satisfies the demand requirements expressed by the traffic trunk's attributes, subject to constraints imposed by resource availability, administrative policy, etc.

A heuristic approach might follow two steps; i) prone resources that do not satisfy the requirements of the traffic trunk attributes; and ii) run a shortest path algorithm for the residual graph. In general, when multiple traffic trunks are to be routed, it cannot be shown that the algorithm always finds a better mapping (or even a solution).

6.3 MPLS-support of DiffServ

Utilising MPLS to carry DiffServ classes has been looked upon with much interest. This is described in e.g. [ID_MPLSdiff]. Then the question arises how to map the behaviour aggregates (BAs) onto LSPs. Basically this can be done by either:

- Using LSPs that carry several Ordered Aggregates (OAs), implying that the Exp field in the MPLS header is used for separating different classes (giving scheduling treatment, precedence, etc.). This is called Exp-inferred PSC LSPs (E-LSPs). Then up to eight BAs (3 bits in the Exp field) can be carried by a single LSP. Mapping from Exp to PHB (PSC and drop precedence) for an LSP is either explicitly signalled or pre-configured; or

- Using LSPs to carry a single OA, saying that the packet treatment can be derived from the Label field while precedence might be derived from the Exp field. This is called Label-only inferred PSC LSPs (L-LSPs). Then an LSP carries a single (FEC, OA) pair. The PSC can be inferred from the label without looking at other information. This implies that the PSC is explicitly signalled at label establishment. The Exp field may give the drop precedence. In case ATM is used, information in the ATM header can be used, e.g. the CLP field.

As mentioned above, combining DiffServ and MPLS allows further differentiation, e.g. by defining different levels of protection for the LSPs.

DiffServ implies service differentiation at every hop, while MPLS and TE may achieve better traffic distribution of the aggregated traffic loads. In this respect, they may work partly independent of each other. Then, MPLS can apply constraint-based routing and admission control for all traffic flows carried in the same LSP. In case more fine-tuning of resources and traffic flows is sought, TE mechanisms may be applied on each service class or smaller groups of traffic flows, e.g. by mapping more specific traffic trunks onto the LSPs.

Some requirements for support of DiffServ by MPLS traffic engineering are listed in [ID_DMreq]:

- Compatibility between mechanisms applied on DiffServ and MPLS level;

- Support of separate constraints on bandwidth for the different classes;

- No additional constraints on the number of class-types and classes;

- Allow for pre-emption per class-type;
- Allow for specifying resource class affinity;
- Support mapping of DiffServ classes as when MPLS is not used;
- Allow dynamic adjustment of PHBs for DiffServ;
- Support multiple TE metrics, e.g. to be used when finding routes of LSPs.

Then a transit LSR can be seen to consist of four functional stages (compare with Figure 23):

1. Determine the incoming PHB (i.e. which PHB the packet belongs to);
2. Determine the outgoing PHB (optionally with traffic conditioning, e.g. policing). When the conditioning is not present the outgoing PHB is equal to the incoming PHB;
3. Label swapping, i.e. from an incoming label to an outgoing label;
4. Coding of DiffServ information into the "encapsulation layer" information, e.g. Exp field, CLP, etc.

Suggestions for mapping between DiffServ classes and "encapsulation layer" information are given in [ID_MPLSdiff].

In order to establish LSPs supporting DiffServ by signalling, the signalling protocol has to be extended. For example, for RSVP a DiffServ object has been defined, see [ID_MPLSdiff]. For an E-LSP this object includes a description of the mapping between Exp values and PHB. This object contains the PSC for an L-LSP. Both E-LSPs and L-LSPs can be established with bandwidth reservation or without reservation. When bandwidth is to be reserved, a TSpec field is included in the PATH message and a Flowspec field is included in the RESV message as described for IntServ/RSVP.

For LDP a new TLV field (DiffServ TLV) is described in [ID_MPLSdiff]. When a predefined mapping is applied between Exp and PHB, use of this field is optional. The DiffServ TLV includes information for mapping between Exp and PHB for an E-LSP. For an L-LSP the DiffServ TLV describes the PSC supported by the LSP. The Label request, Label mapping, Label release and Notification messages may include the DiffServ TLV. Bandwidth reservation may be done by including the Traffic parameters TLV.

7 Resource Reservation

When reserving capacity for a flow (or flow aggregate) a setup protocol can be used. An option is to apply management-related procedures for this purpose, implying that the management system could interact with the routers instead of signalling being exchanged directly between routers. In addition, a combination of management and signalling procedures may also work, for example when combining the action with policy matters, bandwidth brokers, and so forth.

7.1 Resource Reservation Protocol (RSVP)

The Resource reSerVation Protocol (RSVP) is frequently referred to when discussing reservation of resources. The signalling sequence is illustrated in Figure 24. RSVP was designed to enable senders, receivers, and routers of communication sessions (either multicast or unicast) to communicate with each other in order to set up the necessary router state to support the services. RSVP is receiver-oriented, e.g. to overcome scalability problems for multicast and to allow heterogeneity for multicast.

Each RSVP-capable node handles reservation and enforcement of traffic flows by using several modules (see Figure 25). The modules related to Integrated Services/RSVP in routers and hosts are the same, but the actual implementations are commonly different. An RSVP process on both hosts and routers handles all the RSVP protocol messages needed to establish reservations.

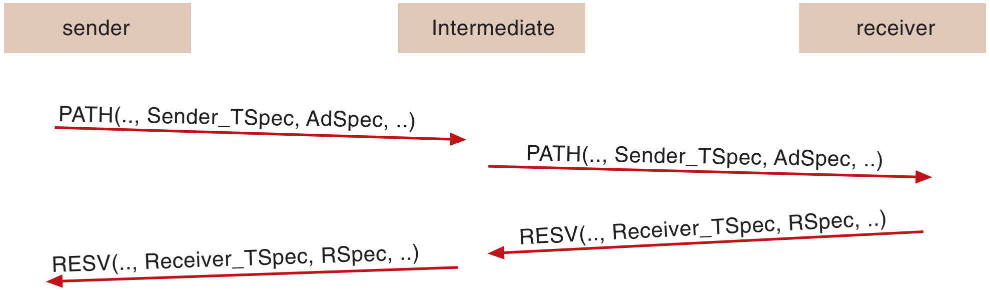

sender
Intermediate
receiver
PATH(.., Sender_TSpec, AdSpec, ..)
PATH(.., Sender_TSpec, AdSpec, ..)
RESV(.., Receiver_TSpec, RSpec, ..)
RESV(.., Receiver_TSpec, RSpec, ..)

Figure 24 Illustration of the RSVP message sequence for setup

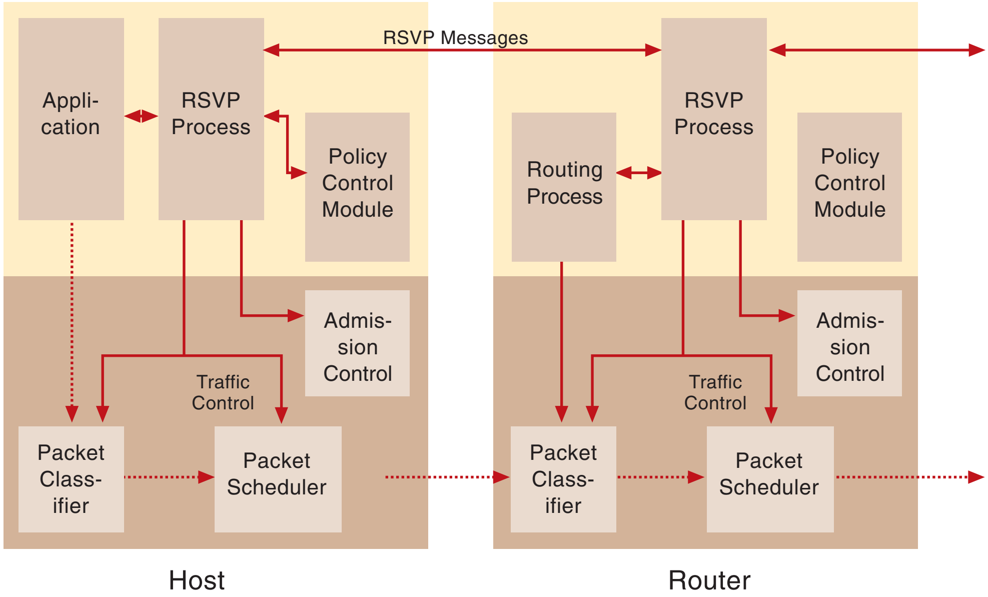


Figure 25 RSVP implementation overview

The different modules are:

- The RSVP process taking care of processing of RSVP PATH and RESV messages.
- The Policy control module being responsible for enforcing policies. That is, the policy control module answers questions like "is the user allowed to do this" (ref. [Jens01a]).
- The admission control module being responsible for ensuring that there are enough resources for the admitted flows. If there is a shortage of resources the admission control module will deny reservation requests.
- The Packet Classifier and Scheduler supporting appropriate handling of the traffic flows. The Packet Classifier looks at every data packet to determine whether the appropriate flow has a reservation and which service class the flow belongs to. The Packet Scheduler then makes the forwarding decision according to the class.
- The RSVP process would also interwork with the Routing process since the routing information is subject to dynamic changes.

RSVP identifies a communication session by the combination of destination address, transport-layer protocol type, and destination port number, as given by the Multi-Field. It is important to note that each RSVP operation only applies to packets of a particular flow; therefore, every RSVP message must include details of the flow to which it applies.

The primary messages used by RSVP are the PATH message, which originates from the traffic sender; and the RESV message, which originates from the traffic receiver. The primary functions of the PATH message are firstly to install reverse routing state in each router along the path, and secondly to provide receivers with information about the characteristics of the sender traffic and end-to-end path so that they can make appropriate reservation requests. The primary function of the RESV messages is to carry reservation requests to the routers along the distribution tree between receivers and senders.

RSVP messages can be transported "raw" within IP packets using protocol number 46, although hosts without this raw input/output (I/O) capability might first encapsulate the RSVP messages within a UDP header.

7.1.1 TE-related Parameters

As described above, the sender initiates a PATH message, which carries a number of fields. Two of these fields are of special interest as seen from a traffic engineering point of view; Sender_TSpec, and AdSpec. The Sender_TSpec field carries information about the traffic to be generated. This information is given as a set of token bucket parameters: token bucket rate [r], token bucket size [b], peak data rate [p], minimum policed unit [m] and maximum packet size [M]. For IntServ, these parameters are given for both Guaranteed and Controlled Load service class, ref. Chapter 5.

The AdSpec field includes flags telling whether or not resources can be reserved along the complete path. These flags are commonly called "break bits" as they indicate if gaps in the RSVP/IntServ scheme were faced by the PATH message. The AdSpec field is put together by fragments; starting by default general parameters, followed by fragments for each function that is selected by the sender application. Absence of a fragment indicates that the sender application does not know or care about that functionality. Then neither the intermediate nodes nor the receiver node should select such functionality. Information in the AdSpec field can be modified by intermediate nodes. Typical fragments in the AdSpec field, in addition to the flags are: number of hops, estimate of bandwidth available, minimum path latency and maximum transfer unit. These are described in [RFC2215].

The combination of Receiver_TSpec and RSpec fields in the RESV message is frequently called the FlowSpec field. This carries the information from the receiver through the network, indicating which functionality and values of parameters that are requested. For the Controlled Load service, the FlowSpec field contains the same set of parameters as for the Sender_TSpec (i.e. mainly the leaky bucket parameters). For the Guaranteed service two more parameters in addition to those in the Sender_TSpec field are given. These are referred to as the RSpec; the rate [R], and the slack term [S]. The RSpec parameters are also described in [RFC2212].

RSVP defines a session to be a traffic flow with a particular destination and transport layer protocol. The RSVP can then be used by end applications to select and invoke the appropriate class and QoS level. It has been claimed, see [RFC2208], that RSVP does not scale to the size of the Internet. To overcome this problem several solutions have been proposed as described in the following subsections.

7.1.2 Class-based Aggregation

One intention of introducing aggregation is that individual flows do not have to be reflected by a state in each intermediate router. The states refer to a class and then the traffic flows are put into the set of classes.

On entering the aggregating region, each flow for which a reservation was made, is assigned to one of the service classes. Flows with similar service requirements are grouped together into a service class. (Service class definition and flow assignment are subjects of ongoing research.) Each packet is marked with a tag that identifies which service the flow should receive. For IP, this tag could consist of the Type of Service (ToS) bits in the packet header (e.g. similar to DiffServ) or the packet could be encapsulated (e.g. similar to MPLS). Inside the aggregating regions, packets are scheduled according to their assigned service class. Because the number of classes is given, packet scheduling is less demanding. However, there is a risk of congestion within any service class. Instead of simply combining all flows blindly into one service class, the overall bandwidth available for each service class can be specified. RSVP-based admission control could be used to admit new flows if there is sufficient bandwidth within the service class. In this manner, the advantages of admission control still apply, but the packets within each service class can be processed and routed more efficiently.

7.1.3 Hierarchical RSVP

Activities within the IETF are also examining hierarchical RSVP. While set-up and release patterns of single RSVP flows are unpredictable, the aggregation of a greater number of flows seems less varying. The idea of hierarchical RSVP is that routers at the edge of aggregating regions use RSVP to reserve large "pipes" with particular characteristics through the region. At the ingress router, packets are assigned to a pipe and encapsulated so they can be classified and scheduled as part of the pipe. Source and destination of the encapsulated packet would be ingress and egress routers. A limited number of different service classes is available. Since RSVP is receiver-oriented, pipe reservations have to be made by egress routers. Egress routers could reserve a number of pipes by default and then adjust the reservations as the actual demand becomes known. When the demand changes, pipe reservations can be further adjusted. A pipe reservation is only maintained if there is a sufficient capacity associated with the reserved flows to use the pipe. Therefore, a router does not necessarily have to have a pipe to every other router, leading to better scaling of the mechanism. Reserved flows for which no pipe exists are served as usual, without aggregation.

The advantage of hierarchical RSVP is the reduction of reservation state information in the routers. Routers in the interior of aggregating regions only keep reservation state for the outer pipe reservations. Packet scheduling is simplified by offering only a few choices of classes. The main disadvantage is that packet classifying is still done by looking into the packet headers and comparing source and destination against the (now shorter) list of reservations.

7.1.4 Enhancing RSVP for MPLS

Once an LSP is established the traffic on the path is identified by the label assigned by the ingress node of the LSP. The packets that are given the same label values by a specific node belong to the same forwarding equivalence class (FEC). When labels are assigned to traffic flows,

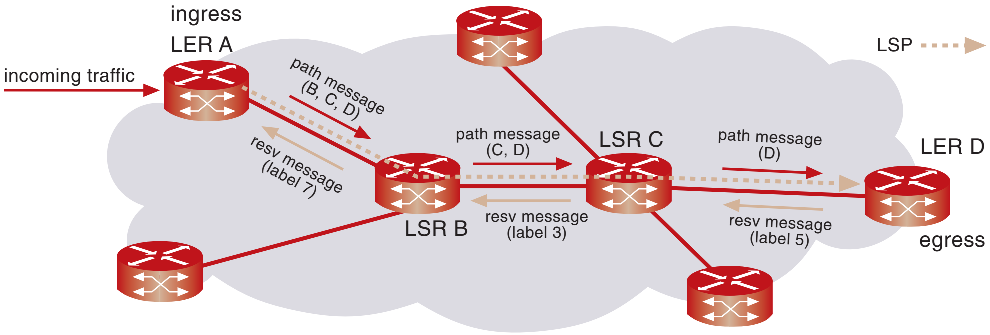


Figure 26 Illustration of LSP set-up by using RSVP

a node may use it to index the corresponding reservation state. Thus, when MPLS and RSVP are combined, the definition of a traffic flow can be made more flexible. Compared to an ordinary RSVP way of identifying a flow, more generality can be obtained when MPLS is also considered. Then, the ingress node of an LSP can use a variety of means to determine which packets that are assigned to a particular label. After assigning a label to a set of packets, the label may identify a flow. The actual packets within the flow are "hidden" for intermediate nodes. Hence these nodes do not need to be aware of which flows (sources, destinations, applications, etc.) that are placed into the LSP.

The setup protocol uses downstream on-demand label distribution. That is, a request introduces an LSP and assigns a label. This is initiated by an ingress node using the RSVP PATH message, see Figure 26. In order to allow this, the PATH message is augmented with a Label_request field. The labels are allocated downstream and distributed (propagated upstream) by the RSVP RESV message (augmented with a Label field). In order to complete the handling of LSPs, procedures for label allocation, distribution, binding and stacking have to be devised. Moreover, the concepts of strict and loose routes and abstract nodes enhance the way of handling LSPs. Five new fields (objects) are introduced in order to handle LSPs with RSVP; Label, Label_request, Explicit_route, Record_route and Session_attribute. In addition, some changes are also seen for the fields Session, Sender_template, Filter_spec and Flowspec.

A key advantage using RSVP to establish LSPs is that allocation of resources along the path is possible. Resource reservation, however, is not mandatory. Such LSPs without resource reservations can be used for example to carry best effort traffic. They can also be used in many other contexts, including implementation of fallback and recovery policies under fault conditions, and so forth.

Using explicitly routed LSPs, a node at the ingress edge of an MPLS domain can control the path through which traffic traverses from itself, through the MPLS domain, to an egress node. Explicit routing can be used to improve the utilisation of network resources and enhance traffic oriented performance characteristics.

Explicitly routed label switched paths can be generalised through the notion of abstract nodes. An abstract node is a group of nodes whose internal topology is opaque to the ingress node of the LSP. An abstract node is said to be simple if it contains only one physical node. Using this concept of abstraction, an explicitly routed LSP can be specified as a sequence of IP prefixes or a sequence of Autonomous Systems.

The signalling protocol model supports the specification of an explicit path as a sequence of strict and loose routes. The combination of abstract nodes and strict and loose routes significantly enhances the flexibility of path definitions.

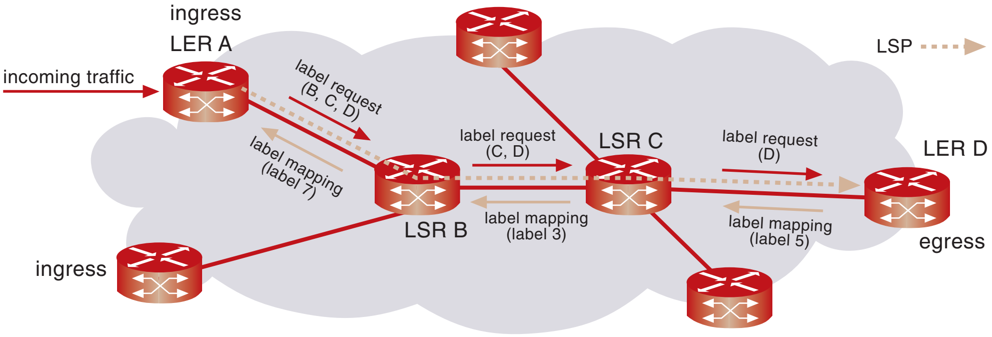
Utilising the combinations of RSVP, MPLS and DiffServ is described in [RFC2430].

7.2 Label Distribution Protocol

The Label Distribution Protocol (LDP) is defined for distribution of labels within an MPLS domain. Hence, RSVP could replace this protocol. Introducing constraint-based routing, Constraint-based Routing LDP (CR-LDP), extends the information used when setting up paths beyond what is available for the routing protocol. The idea is that the LSP will then be better suited to serve the traffic flows. Explicit routing can be said to be a subset of the more general constraint-based routing, as the constraint actually gives the route [ID_crldp].

CR-LDP is a simple, scalable, open, non-proprietary traffic engineering signalling protocol for MPLS IP networks. CR-LDP provides mechanisms for establishing explicitly routed LSPs in an MPLS network, as depicted in Figure 27.

Figure 27 Illustration of LSP set-up by use of CR-LDP



These mechanisms are defined as extensions to LDP. Using CR-LDP, resources can also be reserved along a path to guarantee service levels and adequate handling for traffic carried on the LSP.

To designate an explicit path that satisfies the constraints, it is necessary to discern the resources available to each link or node in the network. For the collection of such resource information, routing protocols can be extended to distribute additional state information.

Additional fields are introduced in the LDP supporting constraint-based routing of LSPs. The following features are supported:

- Strict and loose explicit routing; where the route is given by a list of groups of nodes. In case more than one router is given in the group a certain level of flexibility is present when fulfilling the explicit route.
- Specification of traffic parameters; for instance given by peak rate, committed rate and delay variation allowed.
- Route pinning; which can be used when it is undesirable to change the path followed by the LSP, e.g. in loosely routed segments in case a better route becomes available in that segment.
- LSP pre-emption through set-up/holding priorities; set-up and holding priorities are used to rank existing LSPs (holding priority) and the new LSP (set up priority) to determine whether the new LSP can preempt an existing LSP. Priorities in the range from 0 (highest) to 7 (lowest) are suggested.
- Failure handling.
- LSP identity.
- Resource classes; used when the network resources are categorised into classes to indicate which types of resources an LSP can be placed on (often called colours).

These features are reflected in a number of fields (Type-Length-Value, ref. [Jens01]):

- Explicit route hop TLV – being a series of variable length TLVs where each gives the address of a router (or router group) in a strict or loose sense.
- Explicit route TLV – specifying the path to be taken by the LSP to be established. It is composed of one or more Explicit route hop TLVs.
- Traffic parameters TLV – lists traffic parameters: peak rate (peak token rate, PDR, and maximum token bucket size, PBS), committed rate (committed token rate, CDR, and maximum token bucket size of this rate, CBS), excess burst size, EBS. As seen, a dual token bucket may be used, one operating on the peak rate and another operating on the committed rate. A flag field is used to indicate which of the parameters that can be negotiated. A weight field is also included specifying the LSP’s relative share of a possible excess bandwidth above its committed rate.
- Pre-emption TLV – containing the set-up and holding priorities.
- LSPID TLV – giving the unique identifier of the LSP, composed of the ingress LSR identity (or its IP address) and a locally unique LSP identity for that LSR.
- Resource class (colour) TLV – specifying which link types that are acceptable for the LSP given as a bit mask (32 bits).
- Route pinning TLV – indicating whether route pinning is requested (bit set) or not (bit cleared). A single bit is currently defined.

Category	CR-LDP	RSVP
Transport mechanism	Transport on TCP (reliable)	Raw IP packets (unreliable)
State management	Hard state	Soft state; needs per-flow refresh management
Messages required for LSP set-up and maintenance	Request, Mapping	Path, Resv, ResvConf
Base architecture	Based on LDP developed for MPLS	Based on RSVP, but may require major changes to the basic protocol to improve its scalability
Signalling of QoS and traffic parameters	Can signal DiffServ and ATM traffic classes	Extendable; currently based on IntServ traffic classes
Types of LSPs	Strict, loose and loose pinned	Strict and loose, no pinning
Modes of label distribution and LSP set-up	Easy to support all modes since CR-LDP is based on LDP	Only downstream on demand; need to run both RSVP and LPD for other modes
Path preemption	Supported	Supported
Failure notification	Reliable procedure	Unreliable procedure
Failure recovery	Global and local repair	Global and local repair; local repair done using fast-reroute which requires precomputing alternative paths at every node
Loop detection/prevention	LDP employs Path Vector TLV to prevent Label Request messages from looping. Hop Count TLV is used to find looping LSPs	May be done using the Record Route object
Path optimisation and rerouting	LSP id can be used to prevent double booking of bandwidth for an LSP when doing 'make-before-break'	Shared explicit filter prevents double booking of bandwidth for an LSP when doing 'make-before-break'

Table 2 Comparison of CR-LDP and RSVP (from [Ghan99])

7.3 RSVP and LDP Overview

As described above, both RSVP and LDP can be used for establishing LSPs. These protocols are somewhat different as they were developed for different purposes. RSVP was developed to support the IntServ service architecture enabling an application to signal a request to reserve resources in the network. On the other hand, LDP is developed for the particular purpose of establishing LSPs in the network, i.e. with this protocol information can be exchanged between routers about which labels can be used and for what purposes.

A comparison of Constraint-Based LSP set-up using LDP (CR-LDP) and LSP set-up using RSVP is given in Table 2.

The main differences between CR-LSP and RSVP are that:

- CR-LDP uses TCP whereas RSVP is carried directly on IP (or within UDP), which may imply that information exchange with CR-LDP could be more reliable;
- The direction for reserving resources differs (i.e. ingress to egress and egress to ingress);
- RSVP uses refresh messages for each LSP (some work is taken on to change this);
- CR-LDP might have more problems dealing with failures and requires the rebuilding of LSPs on a backup system. RSVP includes mechanisms to recover from failures, possibly being more fault-tolerant;
- Extensions to RSVP for policy management have been proposed.

8 Concluding Remarks

The main IP-related mechanisms have been outlined in this paper. These are promoted to support a range of services as well as allowing for ensured service levels (at least to some extent). Therefore, most providers are investigating which mechanisms to introduce and how to configure the mechanisms. Another aspect is that different mechanisms may be preferable in different portions of the network/system. However, still the end-to-end service is not adequately delivered. This requires that solutions for mapping between the different mechanisms are in place.

Several of the papers in this issue of *Telektronikk* address various aspects of this. The material in this article is intended to support the basic understanding and thereby ease the comprehension of the remaining material.

References

[Bern00] Bernet, Y. 2000. The Role of the Host Supporting the Full Service QoS Enabled Network. *Internet2 workshop.* Houston. (2001, October 24) [online] – URL: http://www.internet2.edu/qos/houston2000/proceedings/Bernet/20000210-QoS2000-Bernet.pdf

[Bern00a] Bernet, Y. 2000. The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network. *IEEE Com. Mag.*, 38 (2), 154–162.

[COST257] COST. 2000. *Impacts of New Services on the Architecture and Performance of Broadband Networks.* COST 257 Final Report.

[Feng01] Feng, B et al. 2001. State-of-the-art of IP Routing. *Telektronikk*, 97 (2/3), 130–144. (This issue.)

[Ghan99] Ghanwani et al. 1999. Traffic Engineering Standards in IP Networks Using MPLS. *IEEE Com Mag.,* 37 (12), 49–53.

[Henr01] Henriksen, T, Kåråsen, A-G, Wolland, S. 2001. Modelling the topology of IP networks. *Telektronikk*, 97 (2/3), 213–229. (This issue.)

[ID_acaf] Bianchi, G, Blefari-Melazzi, N. 2000. *Per Flow Admission Control over AF PHB Classes.* draft bianci-blefari-admcontr-over-af-phb-00.txt. Work in progress.

[ID_cm] IETF. 2001. Balakrishnana, H, Seshan, S. *The Congestion Manager.* draft-ietf-ecm-cm-02.txt. Work in progress.

[ID_crldp] IETF. 2000. Jamoussi, B et al. *Constraint-Based LSP Setup using LDP.* draft-ietf-mpls-cr-ldp-04.txt. Work in progress.

[ID_DMreq] IETF. 2001. Le Faucheur, F et al. *Requirements for support of Diff-Serv-aware MPLS Traffic Engineering.* draft-ietf-tewg-diff-te-reqts-01.txt. Work in progress.

[ID_dsterms] IETF. 2000. Grossman, D. *New Terminology for Diffserv.* draft-ietf-diffserv-new-terms-03.txt. Work in progress.

[ID_ecn] IETF. 2000. Ramakrishnan, K K et al. *The addition of Explicit Congestion Notification (ECN) to IP.* draft-ietf-tsvwg-ecn-00.txt. Work in progress.

[ID_MPLSdiff] IETF. 2000. Le Faucheur, F et al. *MPLS Support of Differentiated Services.* draft-ietf-mpls-diff-ext-07.txt. Work in progress.

[ID_tcpecn] IETF. 2000. Floyd, S, Ramakrishnan, K K. *TCP with ECN: The treatment of Retransmitted Data Packets.* draft-ietf-tsvwg-tcp-ecn-00.txt.Work in progress.

[Jens01] Jensen, T. 2001. Internet Protocol and transport protocols. *Telektronikk*, 97 (2/3), 20–38. (This issue.)

[Jens01a] Jensen, T. 2001. Traffic Engineering – Inter-domain and Policy Issues. *Telektronikk*, 97 (2/3), 170–185. (This issue.)

[Kell00] Kelly, F P, Key, P B, Zachary, S. 2000. Distributed Admission Control. *IEEE Journal on Selected Areas in Communications,* 18 (12), 2617–2628.

[RFC1633] IETF. 1994. Braden, R, Clark, D, Shenker, S. *Integrated Services in the Internet Architecture: an Overview.* (RFC 16333.)

[RFC2208] IETF. Mankin, A et al. 1997. *Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement. Some Guidelines on Deployment.* (RFC 2208.)

[RFC2211] IETF. Wroclawski, J. 1997. *Specification of the Controlled-Load Network Element Service.* (RFC 2211.)

[RFC2212] IETF. 1997. Shenker, S, Partridge, C, Guerin, R. *Specification of Guaranteed Quality of Service.* (RFC 2212.)

[RFC2215] IETF. 1997. Shenker, S, Wroclawski, J. *General Characterization Parameters for Integrated Service Network Elements.* (RFC 2215.)

[RFC2309] IETF. 1998. Braden, B et al. *Recommendations on Queue Management and Congestion Avoidance in the Internet.* (RFC 2309.)

[RFC2430] IETF. 1998. Li, T, Rekhter, Y. *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE).* (RFC 2430.)

[RFC2474] IETF. 1998. Nichols, K et al. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.* (RFC 2474.)

[RFC2475] IETF. 1998. Blake, S et al. *An Architecture for Differentiated Services.* (RFC 2475.)

[RFC2597] IETF. 1999. Heinanen, J et al. *Assured Forwarding PHB Group.* (RFC 2597.)

[RFC2598] IETF. 1999. Jacobson, V et al. *An Expedited Forwarding PHB.* (RFC 2598.)

[RFC2697] IETF. 1999. Heinanen, J, Guerin, R. *A Single Rate Three Color Marker.* (RFC 2697.)

[RFC2698] IETF. 1999. Heinanen, J, Guerin, R. *A Two Rate Three Color Marker.* (RFC 2698.)

[RFC2702] IETF. 1999. Awduche, D et al. *Requirements for Traffic Engineering Over MPLS.* (RFC 2702.)

[RFC2753] IETF. 2000. Yavatkar, R et al. *A Framework or Policy-based Admission Control.* (RFC 2753.)

[RFC2814] IETF. 2000. Yavatkar, R et al. *SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks.* (RFC 2814.)

[RFC2990] IETF. 2000. Huston, G. *Next Steps for the IP QoS Architecture.* (RFC 2990.)

[RFC2998] IETF. 2000. Bernet, Y et al. *A Framework for Integrated Services Operation over Diffserv Networks.* (RFC 2998.)

[RFC3031] IETF. 2001. Rosen, E et al. *Multi-protocol Label Switching Architecture.* (RFC 3031.)

[Robe01] Roberts, J W. 2001. Traffic Theory and the Internet. *IEEE Com Mag.,* 39 (1), 94–99.

[Tane96] Tanenbaum, A S. *Computer Networks.* Upper Saddle River, NJ, Prentice Hall, 1996.

Planning and Designing IP-based Networks

TERJE JENSEN, METTE RØHNE, INGE SVINNSET,
RIMA VENTURIN AND IRENA GRGIC

Offering a range of services, it is essential for an operator to configure the network such that the performance levels are achieved as expected. Hence, for a multi-service IP-based network the mechanisms available have to be set up to support the service portfolio. Preparing for this, an operator has to have an apparatus in place to estimate the demands and to design the network. These aspects are addressed in this article.

Terje Jensen (39) is Research Manager at Telenor R&D, Kjeller, responsible for co-ordinating projects in the area of QoS and network design. He earned his PhD degree in 1995 from the Norwegian University of Science and Technology. Other activities include performance modelling and analysis, dimensioning and network evolution studies. He was Task Leader in EURESCOM P806-GI.

terje.jensen1@telenor.com

Mette Røhne (35) is Research Scientist at Telenor R&D, Kjeller. Her main activities include applied QoS, network design and techno-economic studies, performed both in international and national projects. She received her PhD degree in 1999 from the Norwegian University of Science and Technology.

mette.rohne@telenor.com

1 Introduction

Facing the steadily growing portfolio of services that an IP-based network may support, it becomes essential for the operator to design the network appropriately. Hence, tools for planning and designing the network are needed. As for "traditional" networks, a number of scopes and settings may be given also for an IP-based network. Some of these are described for the Traffic Engineering taxonomy [Jens01].

A future IP-based network is expected to allow for service differentiation to be efficiently managed. Naturally, this depends on the cost of introducing functionality allowing differentiation compared with the gains that can be achieved. Anyway, running design algorithms would support an operator to fully exploit the potential gains. Moreover, design algorithms are needed even when a single class of service is supported.

A further argument for executing design algorithms is to find closer estimates for capacity needed and tuning of traffic flow handling, and thereby saving investments. Still the ensured service levels stated in any Service Level Agreements (SLAs) should be fulfilled. Even when no strict guarantees are given, the users do have certain tolerance levels. A central part of carrying out the design is to have estimates of the demand. This implies that both the parameters and their values have to be devised and assessed. As ranges of users and applications appear, devising adequate categories is not a trivial challenge.

Transporting IP packets has mostly been the service delivered by the routers. Then more "advanced" services are supported by a network operator, like address translation and ensured performance levels. For a number of cases separate servers are introduced, sometimes referred to as service handlers. An example is the call handler for supporting telephony in IP-based networks. Capabilities of such servers can also be utilised when identifying the efficient network design. That is, the servers may allow for additional control abilities for handling the traffic flows, like rejecting new flows and recommending routing of flows. Besides support from servers, other mechanisms also have to be defined (e.g. ref. [Jens01a]), such as admission control and policing.

An essential task when designing networks is to introduce a number of logical networks in the same physical network. An example is to have a number of Label Switched Paths (LSPs) as described in [Jens01a]. Each of these LSPs has to be routed, its characteristics defined and the relevant traffic flows mapped onto it. Therefore, a design algorithm has to find which set of LSPs to be set up and how the traffic flows relate to these LSPs.

The main objectives of this paper are to describe inputs and steps for planning and designing IP-based networks, ways of characterising the traffic demands, and an algorithm for designing LSPs in a multi-service network. An overall planning scope is described in Chapter 2. Chapter 3 describes characterisation of applications and their traffic flows. When designing networks, the network building blocks must also be characterised. This is treated in Chapter 4. As network planning and design have been conducted for other networks, a few issues that can be observed and fruitfully utilised for IP-based networks, are described in Chapter 5. Then, Chapter 6 outlines an algorithm for designing a multi-service network. A few examples applying this algorithm are given in Chapter 7. To manage a network, appropriate measurements have to be conducted. Although this is treated in several accompanying papers in this issue of *Telektronikk*, a few complementary topics are also mentioned in Chapter 8.

2 An Overall Picture

2.1 Inputs and Results

When investigating approaches for supporting services there are several aspects that have to be looked into. An overview of the main groups of inputs for such deployment studies is sketched in Figure 1. Here, a network is to be established or changed to support the services. Hence, a set of

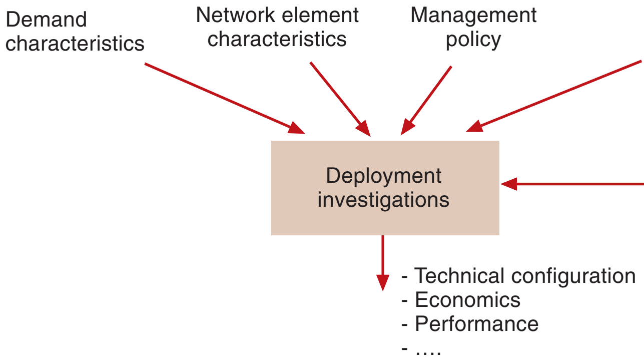

Demand characteristics
Network element characteristics
Management policy
External factors
Deployment investigations
Charging and accounting policy
- Technical configuration
- Economics
- Performance
- ….

Figure 1 Groups of input data for service deployment investigations

Inge Svinnset (46) is Senior Research Scientist at Telenor R&D. His research interests include teletraffic, network performance, Quality of Service and dimensioning. During the last ten years he has been involved in several European research projects related to ATM network performance and dimensioning. He is currently working with Quality of Service and Traffic Engineering in IP networks.

inge-einar.svinnset@telenor.com

Rima Venturin (32) is Research Scientist at Telenor R&D. She is involved in projects related to broadband network planning. She is currently working with techno-economics of IP-based networks. Other activities include demands and application modelling. She received her MScEE degree from the University of Zagreb in 1995.

rima.venturin@telenor.com

activities has to be conducted to prepare for provision of services.

These inputs are categorised as:

- Demand characteristics. The demand patterns for the different applications have to be specified. Besides the demand patterns for the applications, characteristics of each application have to be more closely specified, e.g. in terms of required bit rates, delay requirements, etc. This also includes the set of traffic matrices referring to certain instances of time. Commonly a number of traffic matrices would be needed, each related to a set of applications. A traffic matrix gives the amount of traffic requested from an originating location area towards a destination. When the traffic sources are moving, the spatial impacts have to be taken into account more closely.

- Network element characteristics. The network elements belong to the set of building blocks that the network may be composed of. These elements have to be described in ways relevant for traffic and resource handling. That is, information like capacity per unit, hierarchy of units, dependability and load sharing features, queueing management principles, admission control mechanisms, and so forth, have to be specified. When a cost model is used, some of the network element characteristics will also be included in the cost calculations. Which elements to consider depends on the scope of the study. In some cases only routers and transmission link capacity are looked at, while other types (e.g. call servers, management systems, etc.) may also be relevant for other studies.

- Management policy. The main principles of handling traffic and network resources belong to the management policy. That is, principles like which routing policy to apply, which dependability principle to use, and so forth, are placed in this category. Ways of integrating and segregating types of traffic flows may be candidates to be decided on.

- External factors. Other phenomena to be taken into account are put into a common group. Examples of such factors are ways of interconnecting, regulatory directives, competitors' actions, etc.

- Charging and accounting policy. The charging principles may be identified as flat rate charging, volume based charging, time based charging and congestion-based charging, or a combination of these. In addition to the charging principles the tariffs may vary over certain time periods, i.e. over the day and specific days during a week, and so forth.

Deciding upon an efficient network design implies that an objective function has to be specified in order to settle which designs are the better ones. That is, the objective function would be a measure on how good the solutions is, implying that any improvement in the design results in a better objective value. Thus, this can be seen as an optimisation problem using the objective function and a set of constraints. One set of classical objective functions used contains a measure of cost. This will reflect the capacities needed of the different equipment types. The constraints would then state the demands to be served as well as other requirements, e.g. those belonging to the management policy group of inputs.

After finishing the deployment investigations, an appropriate way of configuring the network resources in order to handle the traffic flows is found. In addition to the technical solution, other aspects might also be relevant, like some economic measures.

2.2 Relations with Economic Considerations

Usually, a management team considers the economic variables when deciding upon steps for changing a network. This means that these variables should be estimated. In order to carry out such studies, a combination of technical and economic considerations is needed. That is, the technical aspects may involve demand estimation, description of network building blocks,

Irena Grgic (30) is Research Scientist at Telenor R&D, Kjeller. She is mainly involved in activities related to QoS and charging for different networks and systems, and studies related to network evolution, both in international and national projects. She holds an MSc in Electrical Engineering from the University of Zagreb in 1999.

irena.grgic@telenor.com

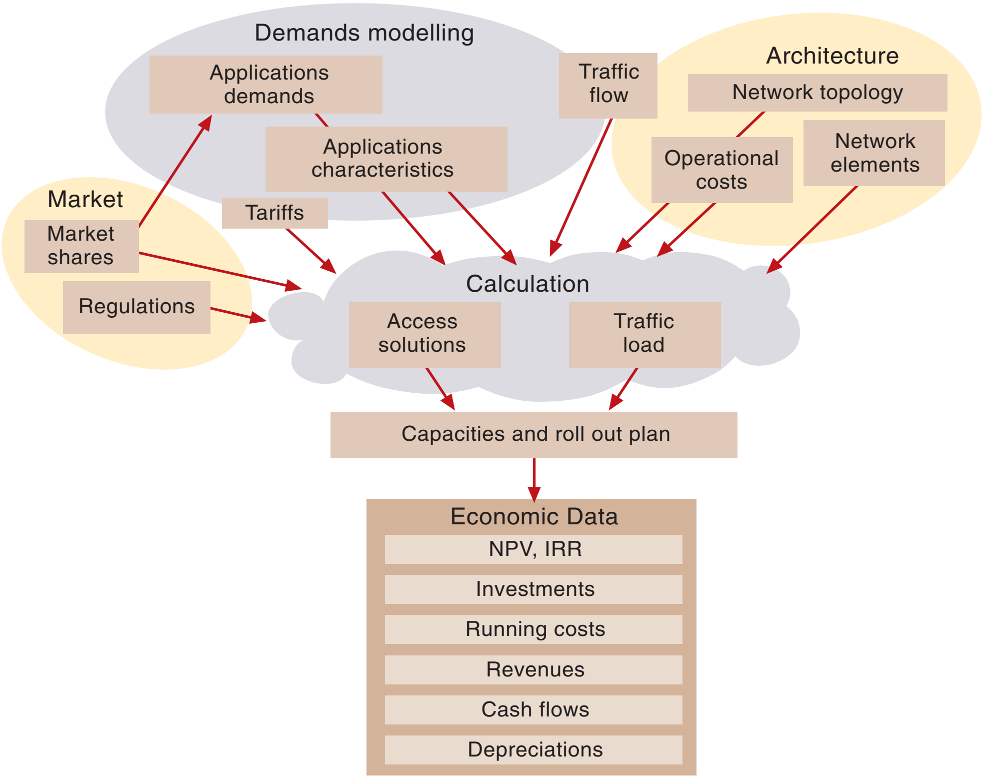


Figure 2 Illustration of work flow for techno-economic studies

technical performance requirements, and so forth. Economic aspects may include requirements on net present value, cash flow restriction, financial conditions, etc. Figure 2 contains an illustration similar to the one in Figure 1, with more emphasis placed on the economic side, on cost and revenue in particular. In order to arrive at a tractable model, quite a few approximations are made on the technical side.

A techno-economic study will capture a number of time periods (e.g. years). The variables must then refer to a number of the time periods, or an evolution of the variables must be given. Two examples of traffic load are given in Figure 3. Starting with traffic load for individual applications, the total traffic load expected from a user can be estimated by looking at the simultaneous use of the applications. Commonly a set of reference periods is assumed for these calculations (e.g. similar to some "busy hour" although a shorter period than one hour would likely be assumed). The reference periods may refer to morning, afternoon, evening, or any other practical identification.

Given these loads, the performance targets and the capacities of the network elements, a roll-out plan for the network arises. An aggregated plan showing only the number of elements is given in Figure 4. Then, having the number of units needed for each year and the cost for each of the units, a cash flow can be obtained. Here the demands may also be related to a set of revenue streams where a mixture of subscription rate, usage rate and income from other parties (e.g. for commercial) may differ for the different applications and user groups.

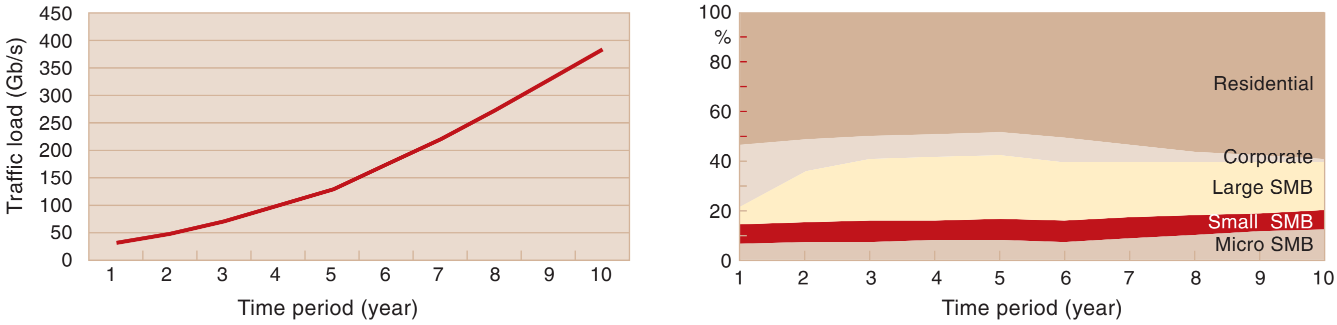


Figure 3 Examples of traffic load inputs for a 10 year study period (left – aggregate traffic load, right – fraction of the traffic load referring to customer segments)

In addition to the roll-out plans, accompanying economic measures are also asked for. Some of these are illustrated in Figure 5 a) and b).

Trying to predict a future situation, most of the values are associated with uncertainties. Therefore, carrying out sensitivity analyses becomes essential. By these analyses one reveals which parameters are significant and which ones have less influence on the overall results. In addition, one may also assess which scenarios are more robust with respect to changes in the input data.

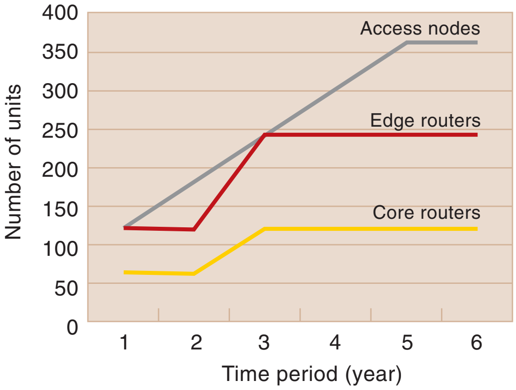
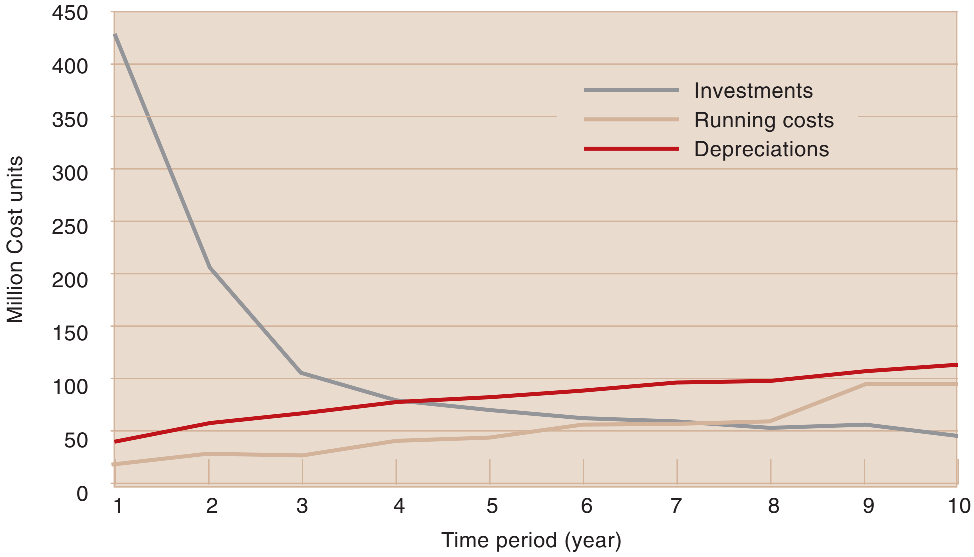
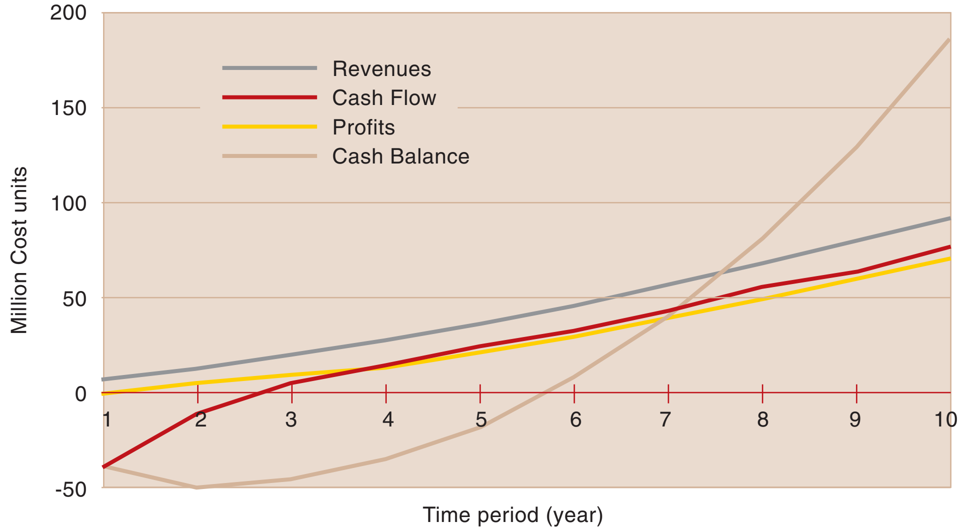


Figure 4 Illustration of a roll-out plan for three types of network elements (in a 6 year study period)

A major concern is that the demand stated by users will likely depend on the conditions they experience. Basically, two types of conditions are recognised – the technical performance levels, and the charges. These are illustrated in Figure 6. The performance level incorporates issues such as effective throughput bit rates, information loss ratios, etc. The tariffs decide the charges facing a user and would thereby influence the interest a user has in invoking services.



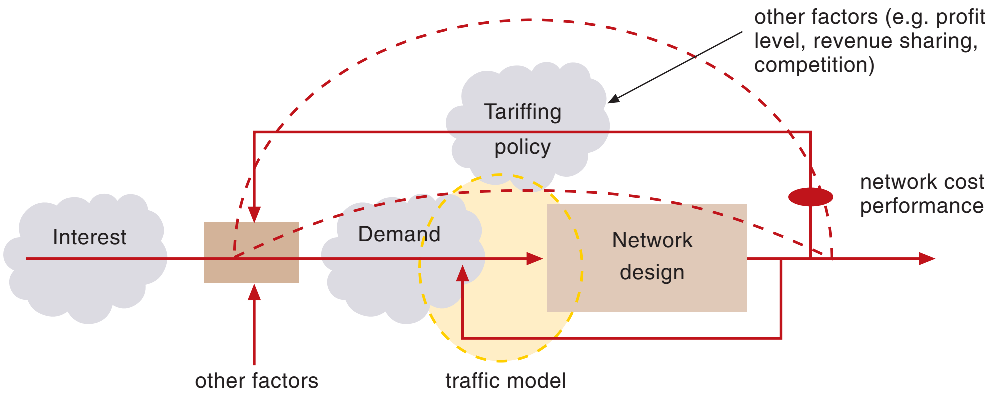
a)

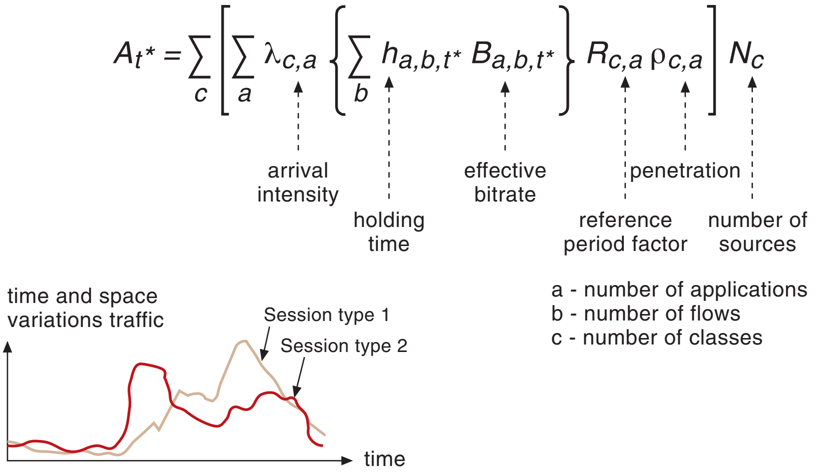


b)

Figure 5 Examples of economic measures for a ten year build-out plan

Figure 6 Feedbacks on two levels – technical and economic





$$A_{t^*} = \sum_{c}\left[\sum_{a}\lambda_{c,a}\left\{\sum_{b}h_{a,b,t^*}\,B_{a,b,t^*}\right\}R_{c,a}\,\rho_{c,a}\right]N_c$$

Figure 7 Multiplying and adding contributions for estimating the overall demands when conducting techno-economic studies

How to efficiently capture these effects in a techno-economic model is a non-trivial challenge. Naturally, the feedback could be included by carrying out iterative algorithms; however, a main question is still how the relations from the factors on the demands really are. For example, how does an increase in the tariff influence the demand for a certain service.

2.3 Estimating Demands – Time Scales and Network Levels

As described above, estimating the demands is conditional for carrying out the network deployment studies. Arriving at a tractable model advocates that several approximations are introduced, leaving the finer details of traffic flow characteristics aside.

For a techno-economic study, average values are likely to suffice. Then a multiplication and addition as illustrated in Figure 7 may apply. It should be noted that this may be carried out in several ways. Moreover, a number of traffic classes may be supported, such that the operations should be done for each traffic class.

The parameters considered are:

- Arrival intensity: giving the number of started sessions per time unit;
- Holding time: giving the duration of the session;
- Effective rate: stating the bit rate for the session (note that a session may contain a number of flows, each with individual holding time and bitrate);
- Reference period factor: reflecting the spreading of the session during the day (see illustration in lower left corner of Figure 7);

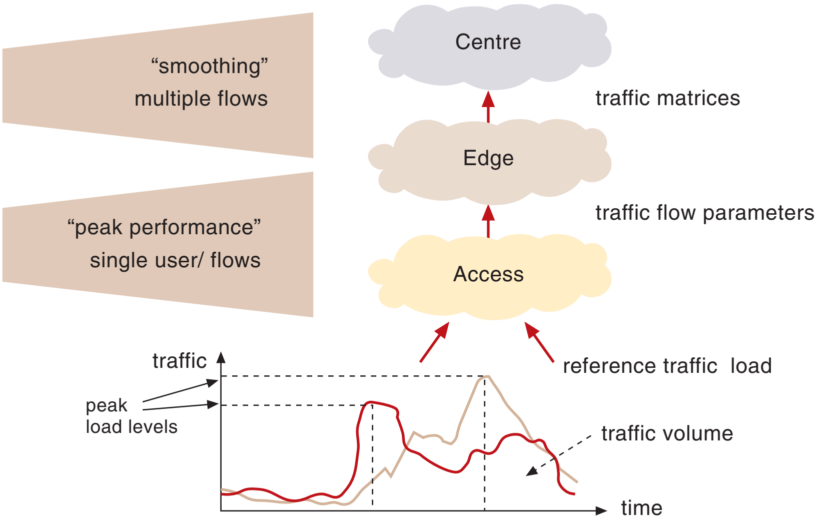


Figure 8 Referring traffic load to network level

- Penetration: giving the fraction of potential users that use the applications;

- Number of sources: giving the number of potential users.

Again, these calculations could be done in a number of ways. Moreover, the values have to be referenced to a certain level in the network, as schematically given in Figure 8.

This is due to the distribution of the traffic, given by the traffic matrices, and the effect that different link rates and other capacity units have on the traffic characteristics. Naturally, this distribution might differ for the different applications, e.g. due to accessing servers that are located at certain sites.

3 Characterising Applications

3.1 Traffic Flows and Service Implementation

A service class refers to a coherent way of treating traffic flows. That is, it includes traffic handling mechanisms/parameters and it may include features like support of multicast, security and mobility. Identifying the proper set of service classes would also be an essential point in the business decisions taken by an actor.

The mapping of traffic flows resulting from the use of applications into the set of service classes may be a non-trivial task. Besides, a service class and accompanying ways of handling the traffic flows may refer to different aggregation levels and portions of the network as described in the previous chapter.

Traffic handling is the set of rules applied by the control and network management system. On this set of rules decisions like how to handle flows in the network can be made. A segregation scheme is applied when the set of network resources is divided for groups of traffic flows. That is, some traffic flows get preference for an amount of resources. Traffic flows with different characteristics are allocated to different Class of Service (CoS) based on various criteria, such as different bitrate demands or different QoS requirements (e.g. delay variation, loss ratio). These may also be considered when CoS indicators are assigned. For some services, other aspects like blocking, control delays and dependability requirements can be examined. The routing scheme can differ for different traffic streams.

Various factors may influence the resulting traffic load experienced by a network. In several cases feedback effects may be present, like flow control algorithms implemented by TCP and charging schemes. How to reach an efficient interplay with the demand applying such effects is a major area where several questions still remain unanswered. A further example of the technical feedback is illustrated in Figure 6.

In addition to the service characterisation mentioned so far, issues related to implementation could be considered. For instance, three ways of supporting services could be relevant:

i) Service with connectivity to a predefined set of destination points. One example can be a virtual leased capacity service. In this case the Service Level Agreement (SLA) specifies the allowed traffic towards these destination points (pipe model); there are no spatial gambling and the necessary resources can be reserved in the network.

ii) Service with call admission control functionality. IP telephony may be implemented with call admission control deciding whether to accept or reject a given call request based on knowledge about the available resources in the network. If resources are not available, the service is blocked, otherwise the necessary resources can be reserved and the connection established.

iii) A one-to-any service without call admission control functionality. In this case the SLA only controls the volume of the traffic flowing over one user-network interface (of each class and both directions). This is called a hose SLA. The SLA is therefore not enough to control the volume of traffic in a given direction, i.e. a kind of spatial gambling on traffic volume.

The three types provide various means for controlling the traffic flows and resource usage. For some types the result might be less efficient resource utilisation, but then likely accompanied by less complexity.

3.2 Inherent Characteristics of Applications and Traffic Flows

As stated in [Robe01] traffic descriptors should satisfy three requirements:

- Useful for resource allocation;
- Understandable by the user;
- Verifiable at the network ingress.

It is further stated that in practise it is impossible to fulfil all these requirements.

From the outset, two types of traffic flows have been used – elastic flows and inelastic flows. As understood by the former, it can adapt itself to conditions such as network congestion. This is

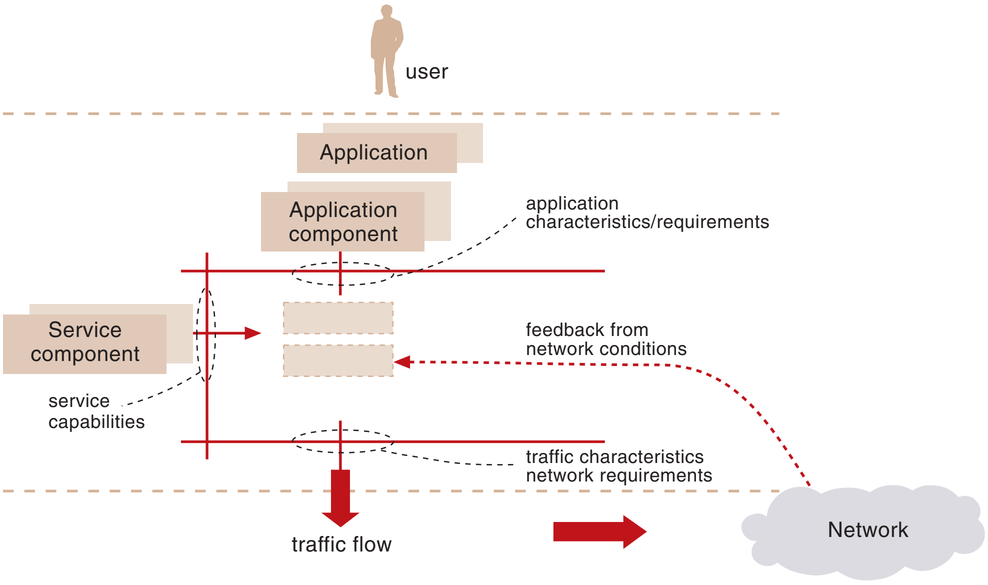

user
Application
Application component
application characteristics/requirements
Service component
feedback from network conditions
service capabilities
traffic characteristics network requirements
traffic flow
Network

Figure 9 Resulting characteristics of traffic flows are largely influenced by a number of factors, such as inherent characteristics of applications, usage of applications, protocols used, conditions in the network

for example done by using the control mechanisms inherent in TCP. On the other hand, UDP as used for some inelastic flow does not contain similar mechanisms. Hence, the set of protocols applied will influence the resulting characteristics. This is schematically illustrated in Figure 9.

Then different classes of traffic flows may be identified in several ways. One approach is the following categories:

- Real-time stream flows: These would have requirements on low delay, low delay variation, low loss ratios, and behaving so that a fixed bandwidth could preferably be allocated. Examples are uncompressed voice (no silence suppression) and constant-rate video.

- Real-time bursty flows: These would have requirements on low delay, low loss ratios and low delay variation, although generating flows with varying bit rates. Examples are compressed voice, variable coded video and shared applications.

- Non-real-time stream flows: These would have requirements on low loss ratios and some requirements on delay and delay variation. Packets will be generated at fixed rate. One example is downloading of video from a server where a play-out buffer is implemented to deal with any delay variation in the network.

- Non-real-time elastic flows: These would have requirements on low loss ratios and some requirements on delay, like when TCP is used and human interaction is involved. One example is web browsing.

- Best effort flows: These would have little requirements, being able to adapt to the network conditions. One example is exchange of e-mails between servers.

A characterisation of the traffic classed for UMTS is summarised in Box A.

When estimating the aggregated traffic it seems to be confirmed [Robe01] that the arrivals of sessions follow closely a Poisson process (due to the inherent nature of superposition of a large number of independent sources). However, the lengths of the sessions may vary (even following a so-called long-tailed distribution). This may be one of the motivations for some applying self-similar modelling (where the traffic flow characteristics look similar on different time scales). The more detailed characteristics of the traffic flows are more relevant when looking at the configuring of units in the network elements, like buffers.

4 Characterising Network Components

4.1 Components of Networks

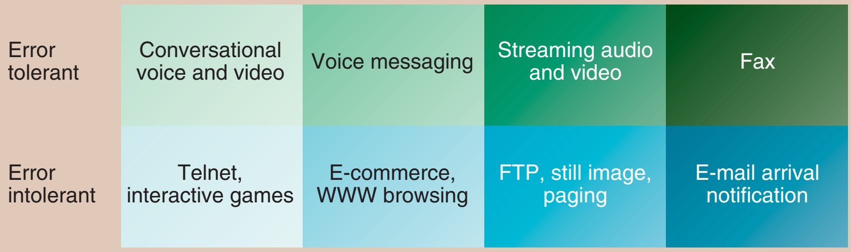
From the outset, different types of resources can be identified in a telecommunication network. Three basic categories are:

- link/transfer bandwidth;
- buffer/storage space;
- computational.

In one way, these resource types may be considered to reflect physical components in a network (e.g. transmission links, RAMs and CPUs, respectively). However, more abstract/logical representation can also be looked at, for instance, when (logical) partitions of a resource

Box A Summarised Traffic Types in UMTS (from [22.105])

The main groups of applications, based on performance requirements are given in Figure A-1.



Error tolerant	Conversational voice and video	Voice messaging	Streaming audio and video	Fax
Error intolerant	Telnet, interactive games	E-commerce, WWW browsing	FTP, still image, paging	E-mail arrival notification

Figure A-1 Main groups of applications (based on performance requirements)

The end-user expectations are given in Tables A-1, A-2 and A-3.

Medium	**Application**	**Degree of symmetry**	**Data rate**	**Key performance parameters and target values**		
				End-to-end one-way delay	Delay variation within a call	Information loss
Audio	Conversational voice	Two-way	4–25 kb/s	< 150 msec preferred < 400 msec limit Note 1	< 1 msec	< 3 % FER
Video	Videophone	Two-way	32–284 kb/s	< 150 msec preferred < 400 msec limit Lip-synch: < 100 msec		< 1% FER
Data	Telemetry - two-way control	Two-way	< 28.8 kb/s	< 250 msec	N/A	Zero
Data	Interactive	Two-way	< 1 kb	< 250 msec	N/A	Zero
Data	Telnet	Two-way (asymmetric)	< 1 kb	< 250 msec	N/A	Zero

Note 1: The overall one-way delay in the mobile network (from UE to PLMN border) is approximately 100 msec.

Table A-1 End user performance expectations – conversational / real-time services

Medium	**Application**	**Degree of symmetry**	**Data rate**	**Key performance parameters and target values**		
				One-way delay	Delay variation	Information loss
Audio	Voice messaging	Primarily one-way	4–13 kb/s	< 1 sec for playback < 2 sec for record	< 1 msec	< 3 % FER
Data	Web-browsing - HTML	Primarily one-way		< 4 sec/page	N/A	Zero
Data	Transaction services – high priority, e.g. e-commerce, ATM	Two-way		< 4 sec	N/A	Zero
Data	E-mail (server access)	Primarily One-way		< 4 sec	N/A	Zero

Table A-2 End user performance expectations – interactive services

Box A continued

Medium	Application	Degree of symmetry	Data rate	Key performance parameters and target values		
				One-way delay	Delay variation	Information loss
Audio	High quality streaming audio	Primarily one-way	32–128 kb/s	< 10 sec for	< 1 msec	< 1 % FER
Video	One-way	One-way	32–384 kb/s	< 10 sec		< 1 % FER
Data	Bulk data transfer/ retrieval	Primarily one-way		< 10 sec	N/A	Zero
Data	Still image	One-way		< 10 sec	N/A	Zero
Data	Telemetry - monitoring	One-way	< 28.8 kb/s	< 10 sec	N/A	Zero

Table A-3 End user performance expectations – streaming services

are considered for a set of traffic flows. Such a logical partition could be a certain amount of transfer bandwidth, or a certain amount of the buffering capacity. Furthermore, a set of resources can be bundled and considered as a component at a certain abstraction level. Such perspectives are likely in a traffic/network management system. These resources are commonly reflected in an objective function used to determine which configuration is the better one. For network design costs of the resources are often essential components in the objective function.

When calculating the overall cost of a network deployment, several aspects should be assessed. These include the routers, the transmission capacity, any service handler, management systems, and so forth. Depending on the scope of the study, some of these may be less relevant. For example, when a network design is to be found, aspects of management systems may not be considered when these are not affected by the resulting solution.

Both hardware and software related costs should be taken into account. In several cases it is seen by an operator that the basic hardware and software come at a specified price, although upgrading the functionality by introducing new software packages would be relatively costly. Such factors would be essential in a techno-economic study, but again might be less relevant during network design.

4.2 A Cost Model

When elaborating a cost model a basic assumption is that sets of resources can be related to certain groups of traffic flows. In particular this is relevant for link/bandwidth capacity. Here LSPs can be applied, possibly with capacity reservation. Some measures of the required capacity are then needed. For simplicity a single measure is adopted, like the effective bandwidth defined. This measure is assumed in the discussion to follow. Other measures might also be incorporated in the model/algorithm presented.

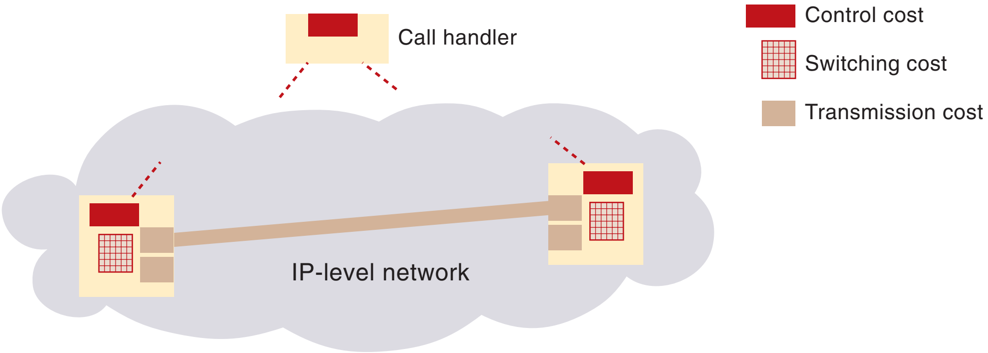


Figure 10 Components of the cost model/objective function

Besides the bandwidth cost, other contributions come from basic router functions (backplane, etc.) and processing in routers and call handlers, see Figure 10. Thus, corresponding contributions have to be incorporated in the cost model. The traffic flows may contribute differently to the various cost components. For instance, a flow not bothering the call handler is likely to have lower cost associated with control (e.g. acceptance control).

Referring to the illustration in Figure 11 three basic contributions to the cost can be identified as described in the following:

Transmission/link/bandwidth cost, Zt_q; reflecting the cost of links between routers as well as termination units within the routers. One unit of transmission capacity between locations *i* and *j* with capacity B_{ij}, has a cost C_{ij}. Each transmission link terminates with a module in location *i* that has a cost C_i. The module may connect N_i of these transmission links. The same relations apply for location *j*. Assuming a traffic aggregate *q* having capacity demand B_q, the bandwidth cost for that aggregate carried from *i* to *j* is calculated as:

$$Zt_q = \frac{B_q}{B_{ij}} C_{ij} + \frac{B_q}{N_i B_{ij}} C_i + \frac{B_q}{N_j B_{ij}} C_j$$

The relation between traffic load in a flow aggregate and the required capacity may not be simple. Again this depends on the service types. For service involving call handler, an effective bandwidth measure may be needed, for instance when doing admission control. For some of these services, a blocking probability measure may well be introduced. Thus, multirate blocking formulae could be applied, abstractly giving the vector of blocking probabilities, $\underline{P_b}$ by:

$$\underline{P_b} = f(\underline{A}, \underline{R}, C)$$

where the load, $\underline{A}$, and the characteristics, $\underline{R}$, of all flows are to be supported by the capacity *C*. A challenge is being able to find the needed capacity, *C*. This could be done iteratively or assuming approximate relationships. In case measurement-based algorithms are used, these parameters may vary. To some extent this could be captured by the effective bandwidth (considering $\underline{A}$ and $\underline{R}$), e.g. see [COST242].

For some "pipe" services a fixed capacity is specified between ingress and egress points (virtual leased capacity service) and this bandwidth value can then be used. Similar arguments might be relevant when a certain level of overbooking is introduced.

The one-to-any service type may have more elastic behaviour, implying that actual relations between traffic load and capacity are not strict. Then a minimum capacity could be obtained assuming some minimum effective bandwidth value (e.g. minimum acceptable throughput).

Switching cost, Zs_q; reflecting the effort requested for transferring packets from an input to an output port. This cost component is assumed to be proportional to the traffic load, A_b, and the effective bandwidth, EB_b, for flows of type *b*. That is, for a traffic aggregate *q* carrying several types:

$$Zs_q = \sum_{b \in q} Zs_b = \sum_{b \in q} A_b(\alpha \cdot EB_b + \beta)$$

where α and β are cost factors. These are chosen to reflect actual router implementations. A central point is to express the effective bandwidth for the traffic flows. For some flows the characteristics may be less influenced by the network load, e.g. for inelastic traffic. Then an effective bandwidth measure could be estimated for the original traffic source characterisation (mean bitrate, peak bitrate, etc.). For other flow types, in particular those using TCP flow control, the resulting characteristics for an individual flow will depend on the network state. Thus, a base estimate could be assumed depending on terminal capabilities, duration of the transfer (due to slow-start behaviour) acceptable delay, etc. An important observation is that some "statistical" effects have to be considered during the network design task. One argument is that several independent sources will be behind the traffic facing the network. Another factor is that several of the values used would frequently be attached with uncertainties, in particular when a future network is to be designed. A more detailed examination of effective rates of such sources could be done when the resulting network design has been found. The switching cost is frequently associated with operations taking place per packet, like exercising traffic handling mechanisms (policer, marker, etc.).

Control cost, Zc_q; reflecting the processing requested for establishing/releasing a connection. Control cost will commonly be related to call handler functionality and other mechanisms implying exchange of information and configuration of relevant traffic handling functions, like policer, marker and shaper. It is frequently assumed that each hop contributes to this cost. That is, if no individual packets but rather aggregates are examined (e.g. due to the use of LSP through-connection) the control cost is likely to be lower for that hop. For a call handler three types of processing steps; connection request, establishment and rejection, are considered.

Assuming there are N paths that could be used for establishing the connection, where path k has blocking probability Pb_k, this cost component could be written as:

$$
\begin{aligned}
Zc_q &= \sum_{b\in q} Zc_b \\
&= \sum_{b\in q} \left\{ \frac{A_b}{h_b} \left[\delta\left(1-Pb_1\right) + \left(N\gamma+\varepsilon\right) \prod_{k=1}^{N} Pb_k \right] \right. \\
&\quad \left. + \sum_{i=1}^{N-1} \left[\left(\delta + i\gamma\right) \prod_{k=1}^{i} Pb_k \left(1-Pb_{i+1}\right) \right] \right\}
\end{aligned}
$$

where connections belonging to aggregate q have a mean offered traffic of A_q with mean duration h_q. The cost factors δ, γ and ε express the effort related to a successful connection, an additional search for paths and rejection of a connection, respectively. As recognised, this expression is adapted from circuit switched networks. Depending on the router mechanisms available, more than one path may not be selectable. Then the expression becomes fairly simple.

Total cost is obtained by adding the different contributions after assigning different weights. In this way the cost for a group of flows, Q, can be calculated as:

$$
Z_Q = \sum_{q\in Q} \left\{ k_t Zt_q + k_s Zs_q + k_c Zc_q \right\}
$$

where k_t, k_s and k_c are relative weight factors related to transmission, switching and control cost. The relative weighting of the cost components is a difficult issue that may have significant implications on the logical network design. It is therefore essential to perform sensitivity analyses varying the weight factors. In principle, these weight factors can be considered as taking part of the cost or placing relative credibility on each, e.g. when certain components are estimated with higher accuracy.

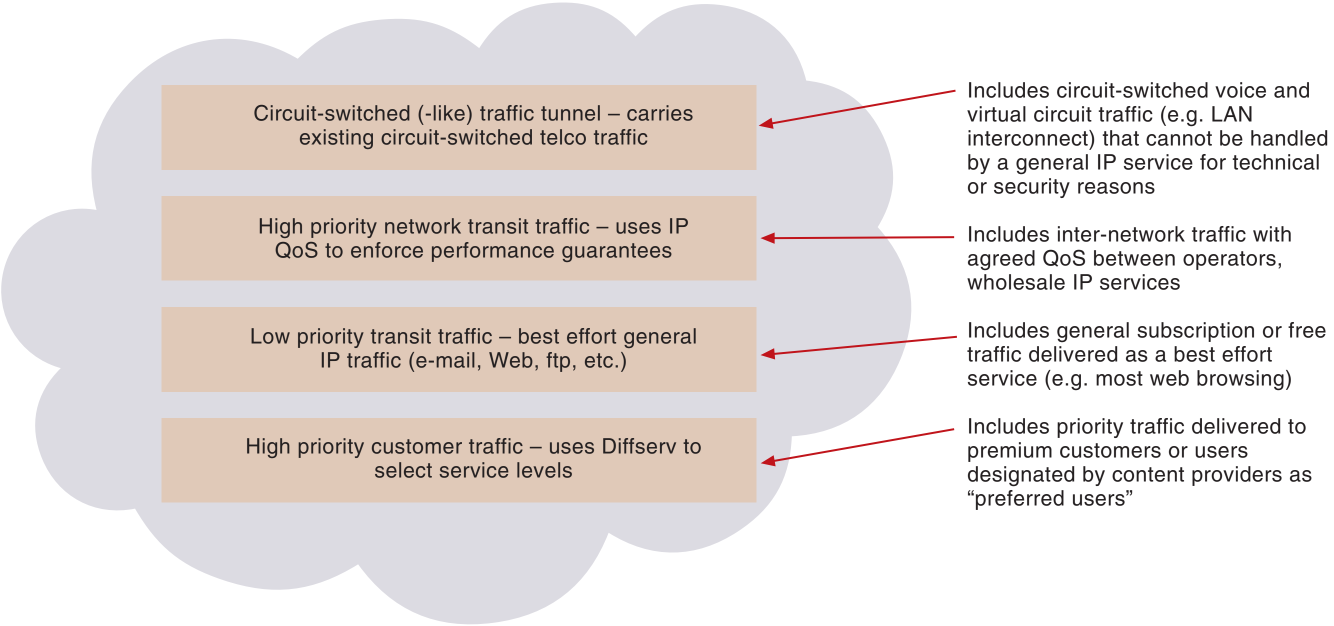
5 Some Lessons Learned Elsewhere

While IP-based networks do have their characteristics, looking at other networks may well give several ideas on how to efficiently configure and manage the network. In most traditional telephone networks, a fixed hierarchical routing has been applied. A number of results have indicated that introducing more dynamic schemes may improve the blocking and the network resilience. Three main types of dynamic routing have been described:

- Time-dependent routing (TDR); changing effective routing tables at pre-planned time instants, e.g. due to daily variations in the traffic pattern;
- State-dependent routing (SDR); changing routing tables according to the network state, given e.g. by traffic load;
- Event-dependent routing (EDR); changing routing tables triggered by certain events, like congestion thresholds crossed.

All these types can be introduced in the routing policies, although the routing protocols would likely take care of the situations that may arise. In one respect, if arrival of a routing message is called an event, the routing could be categorised as event-dependent when looking at an individual router.

Figure 11 A common infrastructure carrying a number of logical networks, adapted from [Ov_NGI]


Circuit-switched (-like) traffic tunnel – carries existing circuit-switched telco traffic
High priority network transit traffic – uses IP QoS to enforce performance guarantees
Low priority transit traffic – best effort general IP traffic (e-mail, Web, ftp, etc.)
High priority customer traffic – uses Diffserv to select service levels
Includes circuit-switched voice and virtual circuit traffic (e.g. LAN interconnect) that cannot be handled by a general IP service for technical or security reasons
Includes inter-network traffic with agreed QoS between operators, wholesale IP services
Includes general subscription or free traffic delivered as a best effort service (e.g. most web browsing)
Includes priority traffic delivered to premium customers or users designated by content providers as “preferred users”

Trade-offs	Traffic management	Routing table management	Capacity management
TE methods applied vs. not applied	TE methods considerably improving performance	Control load comparable for the two cases	Design efficiency comparable for the two cases
Centralised vs. distributed routing table control	Distributed control performance somewhat better (more up-to-date status information	Control load comparable on per-node basis for the two cases	Design efficiency comparable for the two cases
Off-line/pre-planned (TDR) vs. online (SDR, EDR) routing table control	On-line control somewhat better performance	TDR and EDR control load less than SDR	SDR and EDR gives comparable design efficiency, both are better than TDR
FR vs. TDR vs. SDR vs. EDR path selection	EDR/SDR performance better than TDR better than FR	FR/TDR/EDR have lower control load than SDR	EDR/SDR design efficiency better than TDR better than FR
Multilink vs. two-link path selection	Multilink path selection better under overload Two-link path selection better under failure. Two-link path selection lower call set-up delay	Multilink path selection control load generally less than two-link path selection	Multilink design efficiency better than two-link
Sparse logic topology vs. meshed logical topology	Sparse topology better under overload. Meshed topology better under failure	Sparse topology control load generally less than meshed topology	Sparse topology design efficiency somewhat better than multi-area
Local status information vs. global status information	Local status performance somewhat better than global (more up-to-date information)	Local status control load less than global status control load	Design efficiency comparable for the two cases
Status dissemination: status flooding vs. distributed query-for-status vs. centralised status	Distributed query-for-status somewhat better than status flooding and centralized status (more up-to-date information)	Centralized and distributed query-for-status comparable on per-node basis. Status flooding considerably higher control load	Design efficiency comparable for the two cases
Per-flow vs. per virtual-network (per-traffic trunk) traffic management	Comparable performance	Per-virtual-network control load less than per-flow control load	Per-flow design efficiency somewhat better than per-virtual network
Integrated vs. separated voice and data network	Integrated network performance better than separated network performance	Total control load comparable for the two cases	Integrated network design efficiency better than separate network

Table 1 Trade-offs for introducing TE-related mechanisms (from the E.TE series of recommendations)

In case the IP-based network is placed over another network, an overlay model is seen. Then, the underlying network, e.g. WDM or ATM, can be managed separately. However, some initiatives are initiated to see the IP-layer and an underlying network co-operating, like IP over optics.

Traffic Engineering encompasses mechanisms that control a network's response to traffic demands and events that affect the traffic carrying capabilities. As mentioned earlier, TE includes:

- Traffic management, e.g. by controlling routing of traffic, used to maximise the network performance under varying traffic load patterns;

- Capacity management, e.g. by controlling resource configuration, used to design the network in order to minimise its cost while performance objectives are met.

Besides these, network planning can be said to include planning and deploying node and transport capacity in advance of the traffic changes. Thus, these three activities can be said to interact on three time scales.

Learning from other networks a series of studies are documented in [ID_tems], although several of the results are seen for ISDN-like traffic behaviour. Some of the main observations are:

- Network performance seems to always be improved when TE methods are applied, and commonly a substantial improvement is seen.

- Sparse-topology multilink routing networks provide better overall performance under overload than meshed topology networks, although performance under failure may favour meshed topology with more routing alternatives.

- Using state information as in SDR provides essentially equivalent performance as using EDR. EDR is seen as an important class of TE algorithms, being adaptive and distributed in nature. Moreover, EDR may allow less overhead in terms of exchanging routing information.

- Bandwidth reservation is critical to stable and efficient performance of TE methods, and to ensure proper operation of multiservice bandwidth allocation, protection, and priority treatment. Bandwidth allocation per logical network (Virtual Network) is essentially equivalent to per-flow bandwidth allocation when looking at network performance and efficiency and allows great reduction in routing table management and size. This is also proposed as a trend in [Ov_NGI], ref. Figure 11.

- Single-area flat topologies give better network performance and design efficiencies compared with multi-area hierarchical topologies.

- Resource management is shown to be effective to achieve service differentiation. MPLS bandwidth management and DiffServ queueing priority management are important for ensuring that performance objectives are met under a range of network conditions.

- Dynamic transport routing network design improves network performance in comparison with fixed transport routing for all cases examined in [ID_tems] for normal load patterns, abnormal load patterns and failure situations.

Work on traffic engineering is going on in several projects and standardisation groups. For example, ITU-T formulated one question in study group 2 addressing this topic. Seven draft recommendations are under preparation:

- E.TE1: Framework for Traffic Engineering and QoS Methods for IP-, ATM- and TDM-based Multiservice Networks.

- E.TE2: Traffic Engineering and QoS Methods – Call Routing and Connection Routing Methods.

- E.TE3: Traffic Engineering and QoS Methods – QoS Resource Management Methods.

- E.TE4: Traffic Engineering and QoS Methods – Routing Table Management Methods and Requirements.

- E.TE5: Traffic Engineering and QoS Methods – Transport Routing Methods.

- E.TE6: Traffic Engineering and QoS Methods – Capacity Management Methods.

- E.TE7: Traffic Engineering and QoS Methods – Traffic Engineering Operational Requirements.

Some overall observations/results are captured in Table 1.

6 Network Design Algorithm

The objectives of the algorithm for network dimensioning are to obtain the capacities of routers and link sets in this network and to find the number and paths of LSPs that give the lowest total network cost, including control, switching and transmission costs. Introducing LSP capability, the question of whether or not to cross-connect LSPs arise. Cross-connecting traffic flows is a means of separating the traffic flows from their previous LSPs that terminate in the router and putting them into a new LSP that could be cross-connected in this router.

6.1 Initial Steps

To investigate for a better LSP network, trade-offs between control and switching costs and costs for having separate LSPs should be balanced. Typically, additional costs by introducing more LSPs come from dividing the link set capacities into smaller units (optionally reducing the scale effect). Compared with some other approaches (e.g. [E.737], [COST242], [Røhn97], [Popp00]), the algorithm applied does not use a global optimisation formulation. Rather, the procedure has some resemblance with the decomposition approach in the sense that decisions with respect to traffic handling and capacities are made for each location in sequence, iteratively. A flow chart of the initial steps is depicted in Figure 12.

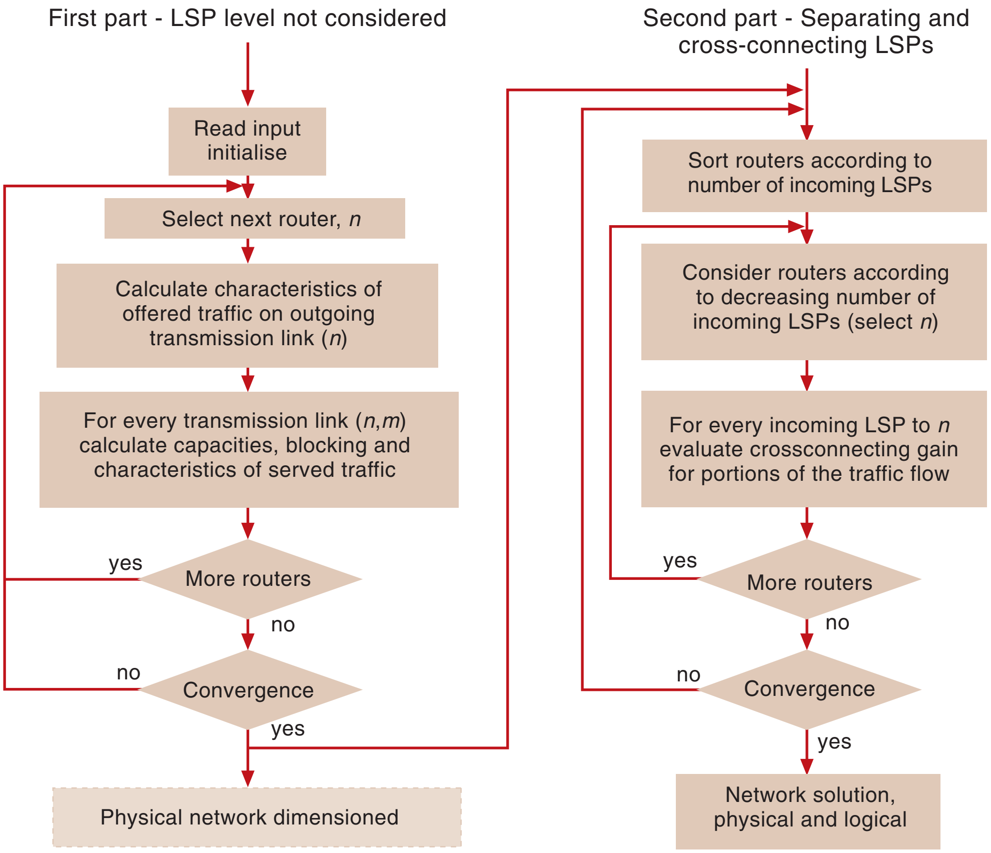

First part - LSP level not considered
Read input initialise
Select next router, n
Calculate characteristics of offered traffic on outgoing transmission link (n)
For every transmission link (n,m) calculate capacities, blocking and characteristics of served traffic
yes
More routers
no
no
Convergence
yes
Physical network dimensioned
Second part - Separating and cross-connecting LSPs
Sort routers according to number of incoming LSPs
Consider routers according to decreasing number of incoming LSPs (select n)
For every incoming LSP to n evaluate crossconnecting gain for portions of the traffic flow
yes
More routers
no
no
Convergence
yes
Network solution, physical and logical

Figure 12 Flow chart showing main outline of program

In the first step the physical network is dimensioned without considering LSPs. Each traffic flow connection will be treated at the IP-packet level in every router it crosses. The iteration is carried out until changes for all the main variables attached to the traffic flows (e.g. mean traffic, capacities needed) are below specified thresholds (convergence criteria). The resulting network from the first step has one LSP per physical link with the capacity of the physical link. It should be noted that an LSP would not be needed and this notion is introduced for simplification.

In the second step we are looking for cost-effective solutions by separating and cross-connecting LSPs. Here, cross-connecting means to extend an LSP through a router (corresponding packets not examined on IP level) and on the subsequent hop (or hops). This is accomplished by looking at one router at the time. The cost model as described above and the relevant segregation schemes are used when deciding whether or not to cross-connect traffic flows by using an LSP. The saving or additional cost related to transmission, switching and control before and after the cross-connection are calculated and compared. If the net saving is above a specified threshold, the bundle of traffic flows will be assigned to an LSP and cross-connected in the router. The segregation scheme is applied when LSPs are considered for cross-connection. Each traffic flow type is characterised by a CoS parameter, and the CoS parameter is used for the segregation. The segregation scheme for LSPs can be defined for any other combination of CoS parameters.

The dimensioning algorithm implemented has a fixed routing scheme and segregates traffic flows according to their CoS parameter. Other schemes for routing and service priority may be considered as well.

The dimensioning procedure results in a cost-effective network solution considering the cost factors and weight factors chosen. The solution has the set of LSPs in the logical network that should be close to the obtainable minimal network cost. The characteristics of the network elements are given as the bandwidth on the physical links of each LSP and the capacity of the routers for switching and control functionality. Variables for the resulting service quality and network utilisation can also be calculated.

The procedure may be used to study the sensitivity for changes in cost factors, weight factors, traffic demand, topology, and so forth. The dimensioning procedure presented is modular. Therefore, several of the steps can be replaced by corresponding expressions such that alterna-

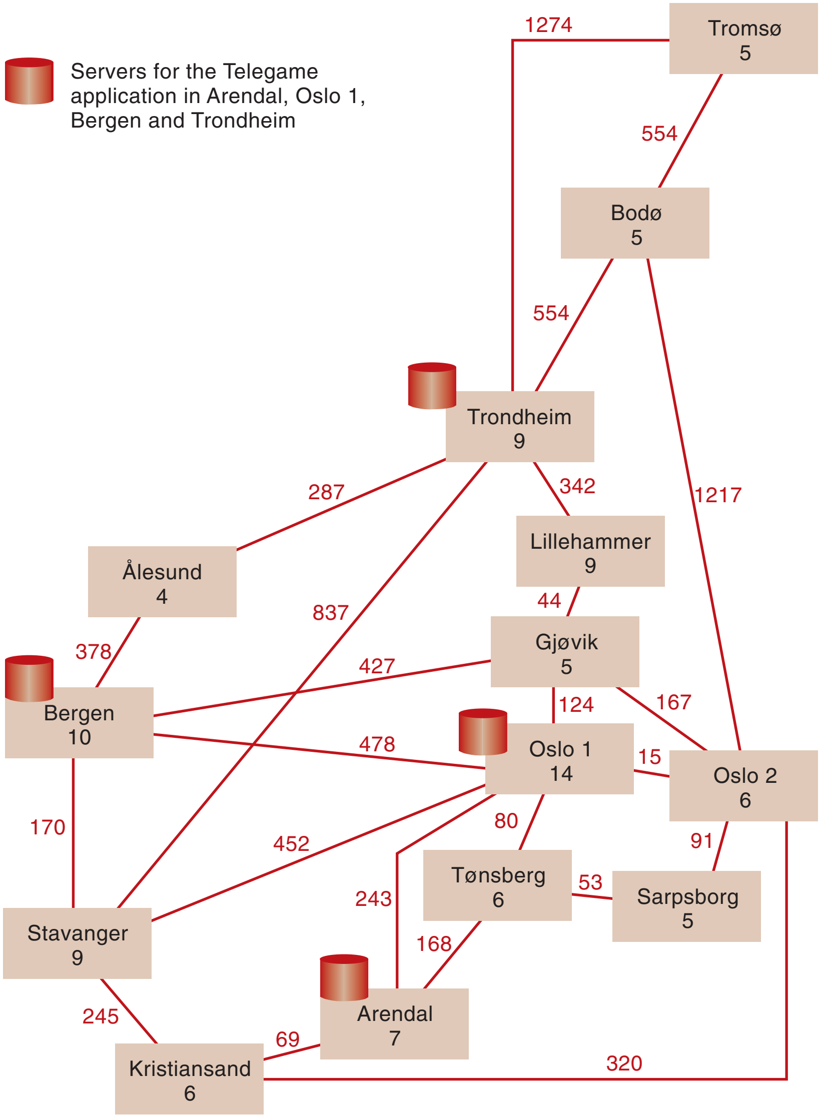


Figure 13 Network structure, giving location identity, relative demand and unit cost per link between locations

tive combinations can be examined. It means that several approximations could be introduced and compared.

6.2 Dependability Concerns

Strategies for establishing protection/backup paths for parts of LSPs and utilisation of such paths may have great implications on dependability and also on the cost of the network design. As seen from an operator's view point, a solution with pre-allocated restoration paths and sharing of restoration capacity between working LSPs seem to be flexible and cost efficient. Then, a set of attributes as described in [Awdu99] can be attached to each LSP, like resilience attribute, pre-emption attribute, adaptive attribute, etc. The backup LSP should be router disjoint from the working LSP and can be established without reserving capacity. This is done as the backup LSP may be common to more than one LSP and these LSPs are likely to have different bandwidth requirements, and the backup LSP pool may be common to many backup LSPs.

In the following it is assumed that two dependability traffic flow classes are used; high and low priority. In case of failure the low priority class may be dropped from the links carrying the backup LSP in case the bandwidth is not sufficient to carry the total load. Each link must then have access to a backup capacity that is at least the maximum of the needed capacity for high priority flows on the working LSPs that will use the link on its backup LSP. A backup capacity may be the difference between the total capacity and the capacity needed to carry the link's high priority traffic flows.

The part of the algorithm related to dependability starts after the initial steps described in the section above. First, all links are considered one by one, and for each link ("failed" link) all LSPs are examined. A backup LSP is identified for the high priority traffic flows having the same end points as the considered LSP on the link. The LSPs on the link are examined according to decreasing cost. As a mixture of low and high priority traffic may flow on an LSP, the needed capacity to restore the high priority traffic is calculated first. Then a route for a backup LSP is found by applying a shortest path algorithm (Dijkstra's algorithm). The "distance" measure used in that algorithm is then the cost of transmission and switching. In case some links having available restoration capacity is looked at the corresponding restoration capacity is attached with zero cost. However, when doing these calculations all needed capacity for high priority traffic on the "failed" link must be taken into account. Finding a backup LSP is only done once for an LSP, meaning that for some LSPs on a link the above calculations may already have been done when that link is examined.

7 Examples

The numerical results given in this section are based on an example network depicted in Figure 13. The network consists of routers that are placed at 14 locations in Norway.

The network structure is described by the following parameters:

- Location identity – given by the name of the city where the router is located;
- Relative demand – presented by a percentage of the total traffic generated;
- Unit cost per link between two locations.

Five applications are considered and characterised in Table 2. For simplicity all applications are assumed to use a call handler, implying that blocking requirements could be more relevant. The total demand per application is also given

Table 2 Applications with relevant input

Application name							
Flow ID	**Direction**	**Mean rate [kbit/s]**	**Peak rate [kbit/s]**	**Loss ratio**	**Blocking req.**	**Duration [min]**	**CoS**
1 Telephony – total demand 151.446 Erlang							
1.1	UN	64	64	10^{-6}	0.01	5	1
1.2	NU	64	64	10^{-6}	0.01	5	1
2 Video on Demand – total demand 2.840 Erlang							
2.1	UN	8	8	10^{-9}	0.01	90	2
2.2	NU	1664	2064	10^{-9}	0.01	90	2
3 Videoconference – total demand 97.567 Erlang							
3.1	NU	8	2000	10^{-6}	0.005	45	3
3.2	UN	1	64	10^{-6}	0.005	45	3
3.3	UN	64	64	10^{-9}	0.01	45	2
3.4	UN	384	384	10^{-9}	0.01	45	2
3.5	NU	64	64	10^{-9}	0.01	45	2
3.6	NU	384	384	10^{-9}	0.01	45	2
4 Real-time transaction – total demand 48.502 Erlang							
4.1	UN	64	128	10^{-6}	0.005	2	3
4.2	NU	64	128	10^{-6}	0.005	2	3
5 Telegame – total demand 998 Erlang							
5.1	UN	64	64	10^{-6}	0.05	20	4
5.2	NU	2000	5000	10^{-6}	0.05	20	4

Direction: UN = user → network, NU = network → user; CoS = Class of Service

in Table 2. In order to calculate elements of the traffic matrix, this total demand is multiplied by a size measure (percentage) for the source and destination location indicated in Figure 13. For example, the element in the traffic matrix for the Videoconference application from Bodø to Bergen is found as: 97567 · 5/100 · 10/100.

Links of capacity 155 Mbit/s are considered. The distance between two locations are multiplied by 100 in order to get the cost for having one 155 Mbit/s link between those locations. In addition, termination modules in the routers represent the other cost component depending on capacity cost. One termination unit able to handle one link is assumed to have cost equal to 10,000 cost units.

The reference values of the cost factors used when deciding whether or not to cross-connect an LSP are given in Table 3. In this chapter, the term LSP is used for any grouping of capacity allocated to traffic aggregates, even though such a grouping may have a capacity higher than a single link (i.e. greater than 155 Mbit/s for these cases). For the calculations presented weight factors for control and switching are equal, $k_c = k_s = k$.

7.1 Reference Case

In the reference case, the configuration and values are kept as described in Figure 13 and in Tables 1 and 2. The network consists of 50 unidirectional links giving a minimum of 50 LSPs. Although these LSPs may not be established, this notion of counting is used for simplification as the relative increase in the number of LSPs and which additional LSPs are established is more interesting than the absolute number of LSPs. Since LSPs on direct links will not be considered for splitting between different CoS classes, no more than 448 LSPs can be established.

The weight factors for control and switching costs are varied and the resulting number of

Table 3 Cost factors and weight factors for reference case

k_t	k_s	α	β	k_c	δ	γ	ε
1.0	k	200.0	0.0	k	1.0	1.0	1.0

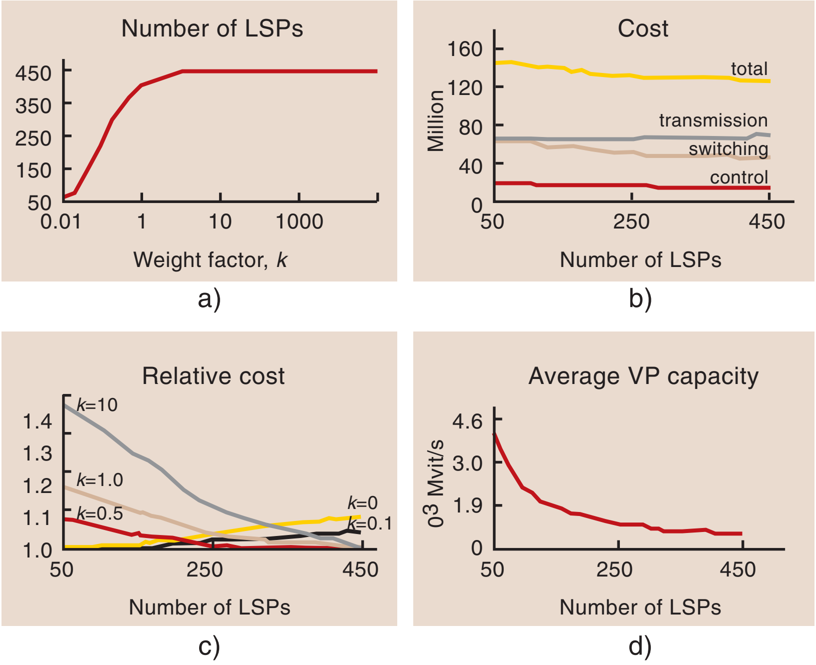


Figure 14 Results for reference case: a) Number of LSPs; b) Network cost; c) Total cost relative to minimum total cost; d) Average LSP capacity as a function of number of LSPs

LSPs that will be established is given in Figure 14 a). Increasing the *k*-factor is the same as increasing the influence from control and switching on network costs. An increase in the costs related to control and switching will make the establishment/cross-connection of LSPs profitable.

The costs related to control, switching and transmission, as well as the total network cost as functions of number of LSPs are given in Figure 14 b). Establishing or cross-connecting LSPs means a splitting of one LSP into two where one of them is cross-connected through the switch. The resulting sum of transmission bandwidth required for these two LSPs will always be greater than or equal to the bandwidth of the LSP that is split. Cross-connecting an LSP implies that less traffic is switched and less control activities would be involved. This leads to lower switching and control cost when the number of LSPs is increased. The minimum total cost for the reference case with the weight factor set to 1.0 is obtained when there are 406 LSPs established. The cost contributions from transmission, switching and control are given, and they seem to counterbalance each other for an increasing number of LSPs in such a way that the total cost is hardly influenced by the number of LSPs established.

A set of curves for the relative total cost for a selected number of control/switching weight factors are depicted in Figure 14 c). The relative costs are found by dividing the total cost obtained by the minimum total cost for the relevant weight factor value *k*. As seen from the curves, a larger number of LSPs are found as the better solutions (minimum relative cost) when greater weight is placed on control/switching cost. The shapes of the curves are explained by this effect. In one respect, these curves show the "goodness" of the solution found compared to alternative solutions. For instance, in case *k* = 1.0, having only 50 LSPs gives a total cost that is approximately 15.5 % greater than the better solution having 406 LSPs.

During the calculations, the bigger LSPs will be cross-connected first. This is clearly shown by the results in Figure 14 d). One rationale for this is that splitting an LSP into two LSPs might lead to less additional need for bandwidth when larger LSPs are considered, as the amount of traffic load (number of micro flows and their characteristics) influences the needed bandwidth. This is recognised as the scale effect observed through the effective bandwidth measure.

7.2 Case Variations

Some variations of the reference case have been examined:

- Single class of service (all traffic flow types are assigned to the same CoS value);
- Higher demand (double demand of reference case);
- Lower demand (one tenth demand of reference case).

Similar results as depicted in Figure 14 can be obtained for these as well, allowing us to identify the LSP network solution with the lowest cost together with some sensitivity results. Figure 15 contains some observations from these case variations.

As seen from Figure 15 a), reducing demands to one tenth, the number of LSPs for the corresponding weight factors is reduced. The flows with reduced demands are smaller than for the reference case, and when splitting the LSPs the relative required capacity for the replacing LSPs is increased, because of the effective bandwidth and call blocking probability.

The total costs of the better network solutions for the four cases are given in Figure 15 b). As expected, increasing the demands leads to higher cost, while reducing the demands leads to lower cost. For the other cases, minor changes to the total cost are found. Related to the costs in Figure 15 b), the relative cost when a minimum number of LSPs and a maximum number of LSPs are established are given in Figure 15 c). To a certain extent, these values indicate the potential savings in finding the appropriate LSPs to be cross-connected for the cases with varying weight factor. The effective bandwidth has a significant dependence on the LSP capacity, and in the case of decreasing the demand the effective bandwidth has a higher relative increase. Therefore the case with reduced demand is more sensitive to the number of LSPs.

For all cases the sum of the required bandwidth on the LSPs on a link is less than the actual link bandwidth, and the link bandwidth not allocated is illustrated in Figure 16 a). When the number of LSPs is increased from the minimum number to the maximum number of LSPs, the required total LSP bandwidth increases. The additional bandwidth caused by establishing LSPs is depicted in Figure 16 b). In these examples, the last type of traffic flow in Table 2 (i.e. Flow id 5.2) has been assigned its mean rate during the capacity calculations. This is to see the effect of less guarantees to the Telegame application. The superfluous capacities on links could be used for additional traffic loads, both giving higher rates to ongoing sessions and potentially accepting more sections (when admission control is used).

8 Measuring Traffic and Performance

8.1 IP Performance Metrics

In order to reach a situation where users and providers of IP services have a harmonised understanding of performance of the network, a set of harmonised IP performance metrics has been devised. The following criteria have been identified to achieve a common understanding, ref. [RFC2330]:

- The metrics must be concrete and well-defined.

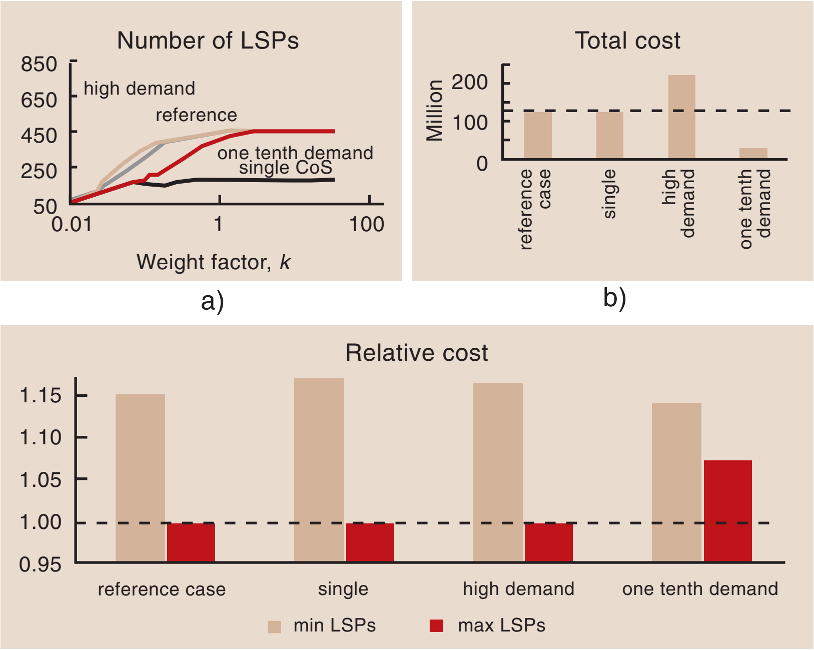
- A methodology for a metric should have the property that it is repeatable (i.e. same results from applying the method several times under identical conditions).

- The metrics must exhibit no bias when identical technology has been used to implement the IP network.

- The metrics must exhibit understood and fair bias for IP networks implemented with non-identical technology.

- The metrics must be useful to users and providers in understanding the performance.

- The metrics must avoid inducing artificial performance goals.

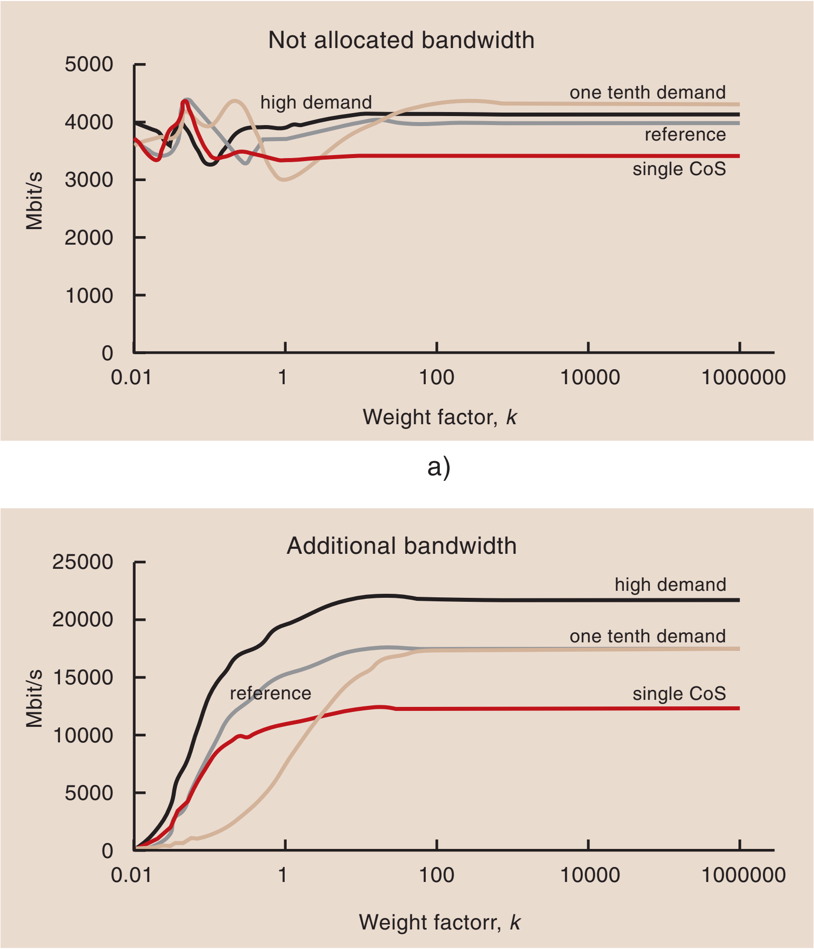


a)

b)

c)

Figure 15 Results from a selection of example variations: a) Number of LSPs; b) Total, k = 1.0; c) Relative cost for minimum number of LSPs and maximum number of LSPs related to the lowest obtainable cost

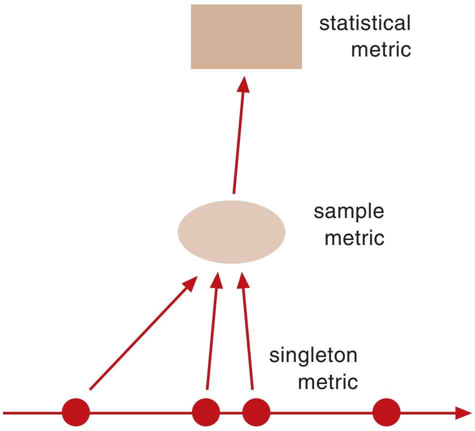


a)

b)

Figure 16 Available bandwidth: a) Link bandwidth not allocated to LSPs; b) additional bandwidth caused by establishing LSPs

Figure 17 Notion of metrics



As identical conditions are commonly quite difficult to achieve, *continuity* is frequently used. Then a methodology for a given metric exhibits continuity if, for small variations in conditions (δ), the variation in the measurements are small (ε). That is, for every positive ε, there exists a positive δ such that if two sets of conditions are within δ of each other, the resulting measurements will be within ε of each other. A metric that has at least one method exhibiting continuity is said itself to exhibit continuity.

Some examples of measurement methods are:

- Direct measurement of a performance metric using injected test traffic. Example: measurement of the round-trip delay of an IP packet of a given size over a given route at a given time.
- Projection of a metric from lower-level measurements. Example: given accurate measurements of propagation delay and bandwidth for each step along a path, projection of the complete delay for the path seen by an IP packet of a given size.
- Estimation of a constituent metric from a set of more aggregated measurements. Example: given accurate measurements of delay for given one-hop path for IP packets of different sizes, estimation of propagation delay of the link of that one-hop path.
- Estimation of a given metric at one time from a set of related metric at other times. Example: given an accurate measurement of flow capacity at a past time, together with a set of accurate delay measurements for that past time and the current time, and given a model of flow dynamics, estimate the flow capacity that would be observed at the current time.

A measurement method is said to be conservative in case the act of measuring does not modify, or has little impact on the value of the performance metric the method is to measure.

When a metric is defined purely in terms of other metrics, it is called a *derived metric*.

A metric can be composed either in a spatial sense or in a temporal sense. The former refers to a case when a metric for a path can be found by considering metrics for subpaths composing the path. The temporal sense refers to a case when a metric for a path at a given time is related to the metric for the path at other instances in time.

Related to measuring, three notions can be used (see Figure 17):

- Singleton metric – a metric that is atomic in a sense (e.g. a single observation);
- Sample metric – metrics derived from a given singleton metric by taking a number of distinct instances together;
- Statistical metric – metrics defined from a sample metric by computing some statistics of the values defined by the singleton metric on the sample (e.g. mean value of a sample).

A way of collecting samples is to undertake measurements separated by certain amounts of time. The time instants can be separated with intervals that are found by sampling from a function, say G(t). If G(t) is a deterministic function, periodic sampling will occur. One major drawback of period sampling is that any periodicity of the traffic flow to be measured may not be easily detected. Therefore, other distribution functions are commonly suggested, like Poisson and geometric.

In [ID_temeas] traffic measurements is interpreted as characterising a flow of IP packets from one point to another. Typical characteristics are (see also [Vike01]):

- Throughput; being a measure of the amount of data passed between two end points. This is commonly given by bits per second or packets per second. In some cases a 5 minute interval is used, allowing for a certain averaging effect at the same time as a so-called active traffic measurement management can be supported. Several values can be given, like mean and 95 % percentile.
- Loss; the loss ratio gives the amount of data not arriving at the far end point divided by the amount of data entering the near end point. Again, measurement interval and ways of

expressing the results have to be decided upon.

- Delay; being a measure of the time taken for a packet to travel from one point to the other. The other point could be the same as the first point for round-trip delay. Interval and result presentations have to be found as above.

- Path; giving the hops that a packet traverses between the end points.

- Lifetime; being the total time that the flow of IP packets exists. For a permanent flow, e.g. a backbone link, the lifetime would be infinite unless there are failures or other changes in the network topology. For dynamic flows, there may be challenges attached to deciding when a flow is started and when it is stopped.

A series of RFCs has been issued for specific performance metrics:

- RFC 2330 Framework for IP Performance Metrics

- RFC 2678 IPPM Metrics for Measuring Connectivity

- RFC 2679 A One-Way Delay Metric for IPPM

- RFC 2680 A One-Way Packet Loss Metric for IPPM

- RFC 2681 A Round-Trip Delay Metric for IPPM

There are multiple purposes of doing measurements including modifying routing for network utilisation, detect threshold crossing for changing capacity allocation, observing trends, observing conditions in SLAs, etc. In particular, measurements are crucial to allow for proactive and real-time TE actions. As one aspect of TE is to reach high utilisation of the network, appropriate load balancing is needed, asking for measurements of the traffic in different directions and on the different links, in order to balance the traffic. Policy-based TE in connection with measurements allow for considering the policy attributes on paths when carrying out actions triggered by measurement results. Such policy attributes include priority, pre-emption, resilience, resource classes and policing.

Performance parameters related to forwarding of IP packets have also been described in ITU-T, e.g. see [Y1540] and [Y1541]. These include IP packet delay variation, IP packet error ratio, IP packet loss ratio, IP packet transfer reference event, IP packet throughput, IP packet transfer delay, and spurious packet ratio.

8.2 Real-time Traffic Flow Measurement

The IETF's Real-time Traffic Flow Measurement (RTFM) Working Group has described a measurement architecture to provide a method for gathering traffic flow information, see [RFC2722]. The model proposed is based on the concepts of meters and traffic flows given as:

- Meters observe packets as they pass by a single point on their way through the network and classify them into certain groups. For each such group a meter will accumulate certain attributes (such as number of packets and bytes). These metered traffic groups may correspond to a user, a host system, a network, a particular transport address (e.g. a port), etc. Meters are placed at measurement points and selectively record network activity as directed by its configuration settings. Meters can also aggregate, transform and further process the recorded activity before the data is stored.

- Traffic flow is said to be a logical entity equivalent to a call or connection. A flow is a portion of traffic that belongs to one of the metered traffic groups mentioned above. Attribute values (source/destination addresses, number of packets, etc.) associated with a flow are aggregate quantities reflecting events that take place. Flows are stored in the meter's flow table.

A traffic meter has a set of rules which specify the flows of interest. One way to identify a flow is by stating values of its address attributes. Annex C in [RFC2722] provides a list of flow attributes.

As well as flows and meters, the traffic model measurement includes managers (to configure and control meters), meter readers (to transport recorded data from meter to analysis applications), and analysis applications (to process the data from meters readings so as to produce whatever reports are required).

NetraMet is an implementation of the RTFM Architecture, which has been available since 1993. Details of the implementation and experience gained with NetraMet are recorded in [RFC2123].

9 Concluding Remarks

This paper has discussed several issues central to carrying out network planning and network design studies. Besides finding efficient algorithms for conducting these studies, input data must also be assessed. Here, as in several other areas, one faces the trade-offs between tractability and accuracy. On the one hand the traffic flows and the network resource could be modelled in great detail, although some problems would likely be faced if these were put together in a larger network.

Inputs and steps for planning and designing IP-based networks have been treated in this paper, including ways of characterising traffic demands and network resources. In addition, an algorithm for designing LSPs in a multi-service network was outlined to show the applicability of the network design.

References

[22.105] ETSI. 2000. *Universal Mobile Telecommunications Systems (UMTS); Service aspects; Services and Service Capabilities.* (ETSI TS 122 105.)

[Awdu99] Awduche, D O. 1999. MPLS and Traffic Engineering in IP Networks. *IEEE Com Mag.,* 37 (12), 42–47.

[COST242] COST. 1996. Roberts, J, Mocci, M, Virtamo, J (eds.). *Broadband Network Teletraffic. Final Report of Action COST-242*. Berlin, Springer Verlag.

[COST257] COST. 2000. *Impacts of New Services on the Architecture and Performance of Broadband Networks.* (COST 257 Final Report.)

[E.737] ITU. 1997. *Dimensioning methods for B-ISDN.* (ITU-T E.737.)

[ID_temeas] Christina, B, Davies, B, Tse, H. 2000. *Operational measurements for Traffic Engineering*. draft-christian-tewg-measurement-00.txt. Work in progress.

[ID_tems] IETF. 2000. Ash, G R. *Traffic Engineering & QoS Methods for IP-, ATM- & TDM-Based Multiservice Networks.* draft-ietf-tewg-qos-routing-00.txt. Work in progress.

[Jens01] Jensen, T. 2001. Traffic Engineering principles, activities and mechanisms. *Telektronikk*, 97 (2/3), 39–53. (This issue.)

[Jens01a] Jensen, T. 2001. Basic IP-related mechanisms. *Telektronikk*, 97 (2/3), 54–85. (This issue.)

[Ov_NGI] Stevenson, I, Pugh, E. 2000. *Next-Generation Internet: Strategies for the Multiservice Network*. Ovum report.

[Popp00] Poppe, F et al. 2000. Choosing the Objectives for Traffic Engineering in IP Backbone Networks Based on Quality-of-Service Requirements. In: *Proceedings of Quality of Future Internet Services (QofIS 2000)*. Berlin, Germany.

[RFC2123] IETF. 1997. Brownlee, N. *Traffic Flow Measurement: Experiences with NeTraMet*. (RFC 2123.)

[RFC2330] IETF. 1998. Paxson, V et al. *Framework for IP Performance Metrics*. (RFC 2330.)

[RFC2722] IETF. 1999. Brownlee, N, Mills, C, Ruth, G. *Traffic Flow Measurement: Architecture*. (RFC 2722.)

[Robe01] Roberts, J W. 2001. Traffic Theory and the Internet. *IEEE Com Mag.,* 39 (1), 94–99.

[Røhn97] Røhne, M. 1997. On the dimensioning of ATM-based VP-networks. In: *Proceedings of the 4th International Conference on Telecommunications, ConTEL97*. Zagreb, Croatia.

[Vike01] Viken, B Å, Emstad, P J. Traffic measurements in IP networks. *Telektronikk*, 97 (2/3), 230–244. (This issue.)

[Y1540] ITU. 1999. *Internet Protocol Communication Service – IP packet transfer and availability performance parameters.* (ITU-T Y.1540.)

[Y1541] ITU. 2000. *Internet Protocol Communication Service – IP Performance and Availability Objectives and Allocations.* (Draft ITU-T Y.1541.)

Achieving Service Differentiation in a Differentiated Services Network by Use of MPLS

INGE SVINNSET

Inge Svinnset (46) is Senior Research Scientist at Telenor R&D. His research interests include teletraffic, network performance, Quality of Service and dimensioning. During the last ten years he has been involved in several European research projects related to ATM network performance and dimensioning. He is currently working with Quality of Service and Traffic Engineering in IP networks.

inge-einar.svinnset@telenor.com

Introduction

In the traditional Internet the only offered service is a Best Effort service. The rapid traffic growth makes the risk of over-provisioning rather low, as what is over-provisioned today will likely be insufficient in the near future. A common opinion today is that IP networks will evolve towards a new advanced architecture supporting also classical Telecom services like voice, and enabling service differentiation providing different levels of performances. This will create new business opportunities for Telecom operators and also accelerate the process of renewing the network infrastructures. To be successful, however, this process requires a greater capability of controlling network performances. As a consequence, there is an increasing interest toward the definition and the implementation of techniques exploitable to meet the desired level of performance under different operational conditions. This new field of activity is usually referred to as Traffic Engineering for IP networks.

Traffic Engineering (TE) is [IETF-TE] a control process or a set of control processes that acts on different time scales with the purpose of performance optimisation of operational networks. In a longer timescale this implies methods for network planning and dimensioning, while in a shorter timescale it implies control aspects of routing and resource allocation.

TE is often treated as virtually synonymous with Multi-Protocol Label Switching (MPLS), but work is ongoing in IETF to provide the necessary features in IP routing protocols for what is known as QoS routing or more generally constraint based routing. On the other hand, MPLS has DiffServ support and is a hot candidate for doing TE in DiffServ networks.

This paper focuses on TE aspects related to techniques that can be used for reconfiguring the network in real-time. A main objective is to be able to provide a consistent set of guidelines for configuring a Differentiated Services IP network based on the available technology, so that different levels of QoS and different levels of service reliability can effectively be met with an efficient utilisation of network resources.

Differentiated Services

Within IETF two IP service architectures have been defined for the purpose of supporting different service demands with regard to network capabilities, Integrated Services (IntServ) and Differentiated Services (DiffServ). The IntServ architecture [RFC1633] relies on the existence of flow specific states that give the possibility to reserve resources end-to-end and in this way realise services with guaranteed performance. The maintenance of these states, however, puts a heavy burden on the IP routers as the state space increases very rapidly, and IntServ is therefore not seen as a scalable solution for future IP networks. But it is still a candidate for the access part of such networks while DiffServ is a major candidate for the core network. This is because DiffServ is believed to be a more scalable way to achieve QoS in an IP network since it acts on aggregated flows and minimises the need for signalling.

The DiffServ architecture is defined in [RFC2475]. It uses a new implementation of the IP version 4 Type of Service (ToS) header octet. This field is now called the DiffServ (DS) field. It has 8 bits, out of which 6 bits are available for current use and two are reserved for future use. The available 6 bits define the DiffServ Code Point (DSCP) and identify a Per Hop Behaviour (PHB). The PHB indicates the way packets shall be handled in the routers and can be set and reset in any DiffServ capable router (marking). Such handling can be delay priorities and drop precedences.

Some PHBs have been standardised; DE (default class [RFC2474]), CS (Class Selector [RFC2474]), EF (Expedited Forwarding [RFC2598]) and AF (Assured Forwarding [RFC2597]). Each AF class uses three DSCP values for differentiating packets with different drop precedences (colouring). This is mainly intended to be used in connection with a congestion avoidance mechanism in the routers in that packets may be dropped based on a given probability that depends on the actual buffer filling and the packets' colour (algorithmic droppers, e.g. Random Early Detection).

An important term is Behaviour Aggregate (BA). This is the aggregation of all packets with the same DSCP crossing a given link in a particular direction [RFC2475]. The set of BAs sharing an ordering constraint is called an Ordered Aggregate (OA). For example, all packets belonging to a given AF class and crossing a given link in a particular direction share an ordering constraint. This is because the AF definition states that AF packets of the same microflow belonging to the same AF class must not be reordered regardless of their drop precedence.

The classification of packets in BAs is based on the Service Level Agreement (SLA) between the parties (e.g. customer and provider). Of interest in this context is the technical part of the agreement, called Service Level Specification (SLS). It contains, among other things, details on how much traffic of different types the customer can initiate and the quality he can expect. From such information the network operator must assure that his network is able to carry all customer created traffic within the contracted limits with a satisfactory quality.

A Per Hop Behaviour Scheduling Class (PSC) is the set of PHBs that are applied to the BAs belonging to an OA [mpls-diff-ext]. For example, the PHBs that are associated with a given AF class constitute a PSC.

Multi Protocol Label Switching (MPLS)

At the IP layer (layer 3) a router makes forwarding decisions for a packet based on information in the IP header. The analysis of the packet header is performed and a routing algorithm is executed in each router. This can be viewed as a two-step process. First the packets are classified into a set of Forwarding Equivalence Classes (FECs). Then each FEC is mapped to a next hop. An FEC is a group of packets that shall be forwarded over the same path with the same forwarding treatment.

With Multi-Protocol Label Switching (MPLS) the classification of packets into FECs is only performed at the ingress to the MPLS domain. The packet is then mapped to a Label Switched Path (LSP) by encapsulation of an MPLS header. The LSP is identified locally by the header, or more correctly by the label field in the header. Based on the value of the label the packet is mapped to the next hop. In successive routers within the MPLS domain the label is swapped (therefore it can have only local significance) and the packet is mapped to the next hop.

The MPLS architecture is described in [RFC3031] whereas support of DiffServ over MPLS networks is described in [mpls-diff-ext].

Class of Service

To be able to design and manage a network for carrying services with different quality requirements, it is necessary to define a set of Classes of Service (CoS) as seen from a network point of view. This set should on the one hand reflect different service and customer requirements, and on the other hand the possibility of the network to provide differentiated service levels. The users must be able to see the difference between the different choices, not only in price but also in service levels.

In [Johnsen 1999] a CoS is defined as *‘a category based on type of users, type of applications, or some other criteria that QoS systems can use to provide differentiated classes of service. The characteristics of the CoS may be appropriate for high throughput traffic, for traffic with a requirement for low latency or simply for Best Effort. The QoS experienced for a particular flow will be dependent on the number and type of other traffic flows admitted to its class.’*

This wide definition opens up for included parameters like packet loss, latency, throughput, as well as survivability aspects in defining the different classes. But it could be discussed whether CoS should be used in a relative sense and the term QoS classes should be used when classes are differentiated with quantitative requirements.

In an MPLS context CoS is used in relation to the CoS-field, which is three bits in the MPLS header. This field can either be used to differentiate between different CoS within an LSP (E-LSP) or to identify colouring in case the LSP is dedicated to one CoS (L-LSP). The classification into CoS is left for the network operators to decide.

In a DiffServ context the term ‘traffic class’ is sometimes used for traffic that shares a common set of QoS requirements. Such a class could be characterised by using a standardised PHB group [diffserv-new-terms] like EF, one of the AF classes or Best Effort (BE), in each node. Such classes are also often referred to as DiffServ classes. CoS in the first definition above could in addition add criteria like resilience so that a DiffServ class could contain many CoS.

An associated term in DiffServ is Per-Domain Behaviour (PDB). This is defined in [diffserv-pdb-def] as “the expected treatment that an identifiable or target group of packets will receive from ‘edge to edge’ of a DS domain”. A particular PHB (or, if applicable, list of PHBs) and traffic conditioning requirements are associated with each PDB.

In this document CoS is used to group traffic on the basis of performance related requirements (quantitative or qualitative) like loss, delay, delay variation, throughput and resilience and in addition priority and elasticity (Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)). A service can consist of one or many service components as illustrated in Figure 1. One example can be a multi-media service with a voice, a video and a data service component. Each service component belongs to a CoS. Finally, a mapping is presumed between CoS and PHB group that is unique within a domain.

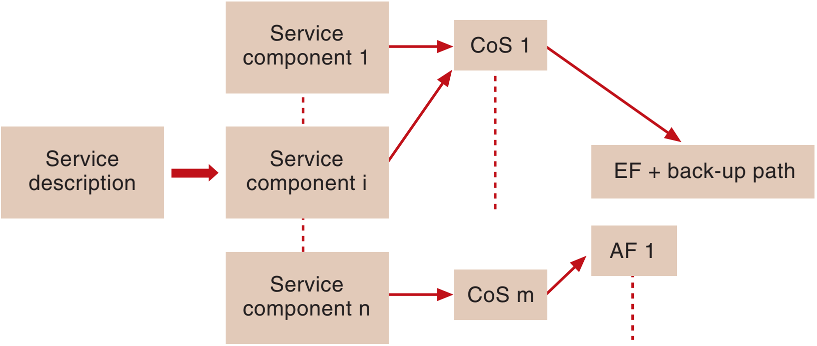


Figure 1 The relation between service description, CoS and DiffServ classes

Lack of Control makes Service Differentiation Difficult

In the following an example is given with traffic offered to a DiffServ capable router (Cisco 7507). The traffic offered is constant UDP traffic from a SmartBits tester. Although this is not a realistic test scenario, since UDP does not have the important feedback control of TCP and normal traffic variations are not present, some important points are shown.

In the test scenario we have used four CoS and Low Latency Queueing towards a POS STM1 interface. The CoS are:

CoS 1: Traffic with strict real-time requirement. The service can be Voice over IP (VoIP). The traffic is mapped to the Expedited Forwarding PHB and uses a strict priority queue with rate limit 23.5 Mbit/s (15 – 16 %). The packet size is 110 byte (IP).

CoS 2: Streaming traffic. This is non-priority traffic with real-time requirements, but with looser requirements than CoS 1. The traffic is mapped to an Assured Forwarding PHB class using WFQ with weight 50 %. The packet size is 942 byte (IP).

CoS 3: Better than Best Effort (BBE) traffic or Business Class. This is non-priority traffic with no real-time requirement but high requirement on loss and throughput. This traffic should be based on a protocol like TCP. The traffic is mapped to an Assured Forwarding PHB class using WFQ with weight 25 %. The packet size is 622 byte (IP).

CoS 4: Traffic with unspecified requirements like today's Internet (but will probably be given higher throughput requirements than in today's network). This is BE traffic and uses WFQ with weight 25 %. The packet size is 622 byte (IP).

Traffic from a given CoS is mapped to a unique queue at the router output interface, and different classes use different queues. While CoS 1 has absolute delay priority over the other classes, the other classes use Class Based Weighted Fair Queuing (Figure 2). So these classed are served according to pre-defined weights.

There is however one exception in this implementation (Cisco 7507) and this is the Tx-buffer. Packets always go via a common buffer, the Tx-buffer. Only when this buffer is full are incoming packets forwarded to the Class Queues as given in Figure 2.

In the example the load is increased linearly so that the relative proportion between the classes is kept. We observe that as the load increases the different classes get their relative share of the throughput as given by the scheduler (fair share). Some classes get more as long as some of the other classes do not use their reserved bandwidth. VoIP starts to lose packets when offered traffic reaches the rate limit configured.

The latency as a function of total offered traffic is given in Figure 3.

In the given example the BE queue starts to grow as soon as congestion state is reached. The latency fast approaches a maximum value corresponding to the buffer size for this queue. Due to the Tx-buffer the latency increases accordingly for all the other classes as soon as congestion state occurs. In the example this increase is from 0.5 ms to 4–5 ms. With the given offered traffic and configuration, VoIP is the next class to get

Figure 2 Low latency queuing

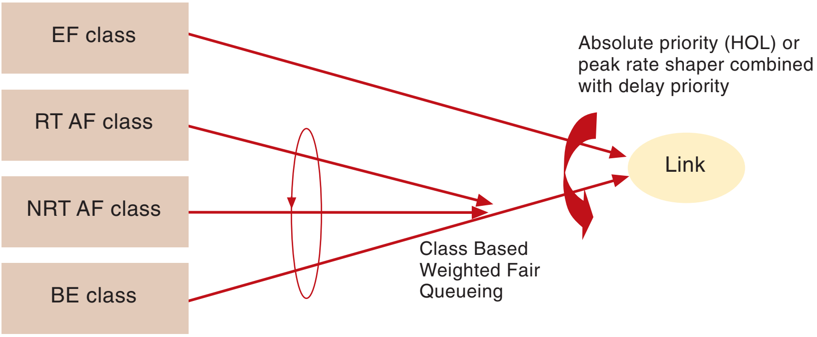
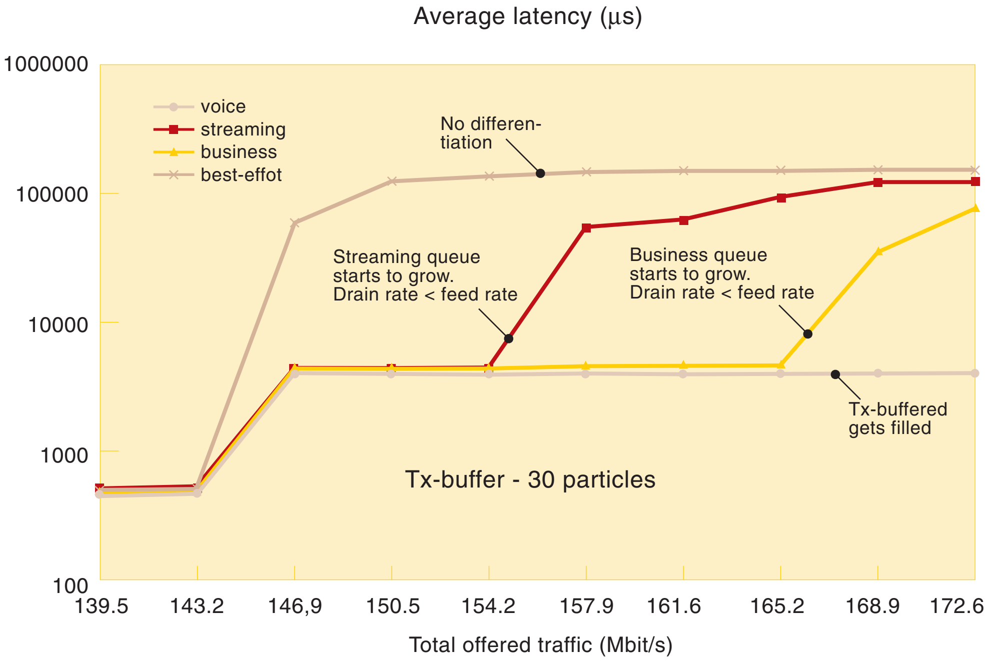

EF class
RT AF class
NRT AF class
BE class
Absolute priority (HOL) or peak rate shaper combined with delay priority
Link
Class Based Weighted Fair Queueing

Figure 3 Average latency distribution



performance degradation. This is due to the policing function (CAR rate-limit), and the effect is packet loss (not shown in the figure). Streaming is then the next class to get performance degradation. As long as offered rate for this queue is lower than the scheduler rate, the average latency is almost as low as for VoIP (difference is 0.4 ms). But this load interval is rather small and when scheduler rate gets too low performance degrades quickly.

Although the example given is not a realistic traffic load scenario, it shows that the load interval where we have some kind of performance differentiation can be rather small and that we need some control mechanism to guarantee performance to some extent. With low load (no congestion) there will be no differentiation between the service classes. Only in case of congestion can we see a difference between the classes. This difference relies on the relation between the relative share of offered traffic and the weight given to the class by the scheduler (drain rate). If the configured rate does not match the actual traffic we may e.g. see that the BE class gets the best performance! Even the performance guarantee of the priority class relies on the fact that offered traffic is below the rate-limit for this class.

A problem with DiffServ is that control of the traffic from a given customer is based on the SLA/SLS that is normally given as a total volume for each class to and from that customer. That is, we can give an upper limit for the traffic (for each class) entering the network, but we do not know how the traffic is distributed. This may create congested points in the network while other parts are under-utilised. Also dimensioning of the network cannot be based solely on the SLS parameters, since these are upper limits and will give an expensive worst case design. A more sophisticated control framework is needed to support a well-dimensioned network that can differentiate between service classes and at the same time give some performance support. This can be based on admission control or on the use of bandwidth reservation on an aggregated level combined with traffic measurements, e.g. by use of MPLS. Also MPLS has support for fast recovery in case of failure that may be required by some services.

The Use of Measurements for Control and Capacity Planning

The utilisation of MPLS simplifies the task of monitoring the traffic on each trunk and building a picture of the load on the network. With the aid of this information it should be possible to

- Manage the network more effectively to obtain better end-to-end performance and more efficient use of resources;
- Guide connection admission control (CAC);
- Build traffic matrixes for capacity planning purposes.

This monitoring should be done at the entrance to the MPLS network, i.e. at the LSP ingress in the edge routers. A clearer picture of the available resources in the network should be gained. By making use of this information in the CAC process, it should be possible to allow higher utilisation without risking congestion.

In addition to flow metering, different other types of QoS and performance measurements and indicators will be important. The most important of these types of measurements will be:

- Measurements of end-to-end delay and the variation of the end-to-end delay;
- Measurements of packet losses in the network;
- Reports about buffer threshold crossings and faults.

These measurements are necessary to control that the QoS requirements for the different traffic classes are fulfilled.

Measured data in a node could be:

- Number of packets and octets;
- Intentional loss (RED, policy, contract enforcement);
- Unintentional loss (buffer overflow);
- Other, e.g. delay statistics through a router.

The integration time for rate measurements depends on the type of traffic we are measuring. The higher the requirement for packet loss / delay performance, the smaller is the window size needed. As a consequence real-time traffic should be measured in shorter intervals than what would be necessary for typical TCP traffic.

Although measurements are believed to be more important in the future and constitute a basic part of the control framework presented in the next section, the most important questions related to how measurements can be performed remain open. Fundamental questions are the cost of implementing monitoring hardware in the routers and possible technology constraints related to monitoring at high speed. What measurement time intervals should in fact be used is also an item for further study.

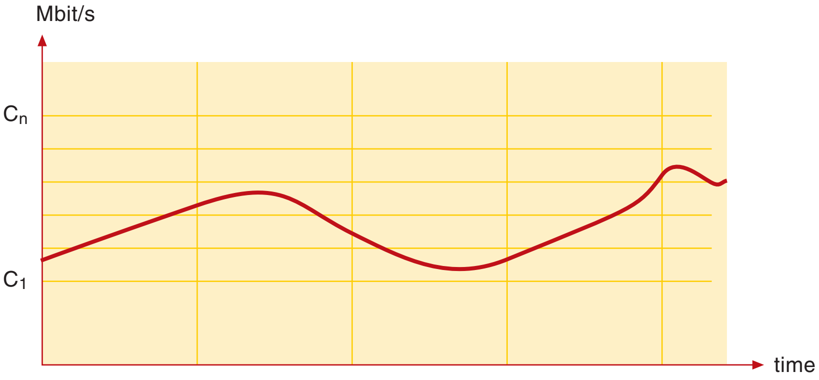
In a network domain deploying MPLS parameter setting for the different LSPs may not be an easy task. Due to the fact that the traffic to be carried on an LSP is more or less unknown it will be difficult to set these parameters in advance based for instance on the SLS. To overcome this problem it will be necessary to monitor the traffic at the edge node of the LSP as discussed above. We assume that the capacity reservation is based on the signalled values at set-up of an LSP and that the traffic entering the LSP is policed according to these parameters. Furthermore we assume that these parameters will be used to divide the capacity among different traffic classes (e.g. to set the different rates or weights in the routers to get the desired traffic performance).

The protocols deployed for setting up LSPs allow different traffic parameters like peak rate and committed rate with associated tolerances to be set. However, due to the fact that the traffic on an LSP will consist of a superposition of traffic from many users it will be nearly impossible to set the appropriate traffic parameters for a given LSP. Generally, it will be difficult to set more than one single bit-rate parameter per LSP. In addition one needs to have a certain tolerance on this bit-rate due to the fact that traffic may stem from many sources and therefore may have unfortunate phasing. Consequently, we propose to base the control framework on the monitoring of one bit-rate parameter Committed Data Rate (CDR) and policed by a bucket with rate CDR and tolerance parameter Committed Burst Size (CBS). Excess Burst Size (EBS) may also be used for elastic traffic (i.e. two buckets).

We assume that the LSP parameters will be updated based on observed threshold crossings according to the following scheme. At set-up a set of initial parameters will be provided. These may be taken as an initial guess based for instance on experience from similar LSPs. Once an LSP is set up the traffic monitoring will be triggered. Based on the measurements performed per LSP, it may be necessary to renegotiate the parameters for the LSPs. Resetting of parameters should be on a rather long time scale (compared to the packet arriving times) to avoid rapid changes and possible instabilities.

Traffic monitoring is a basis for estimation of CDR for a given LSP. In advance we have a set of predefined limiting values for CDR, say C_1, C_2, ..., C_n. The number of levels must be limited to avoid too rapid changes. We assume that the estimate of bit-rate fluctuates rather slowly as a function of time (minutes or hours). This can for instance be achieved by using some kind of weighting.

Figure 4 Estimated traffic as a function of time



As a first approximation, suppose that the estimated bit-rate $\hat{B}_t$ is between C_i and C_{i+1} at time t, then the CDR is set to C_{i+1}. The first time $\hat{B}_t$ crosses one of the levels C_i and C_{i+1} we change the CDR to C_i if the crossing is downwards and to C_{i+2} if the crossing is upwards.

It might also be necessary to introduce some safety factor *safe* so that CDR is set to $safe \cdot C_i$. This could for instance be done for high priority traffic such as the EF class (overbooking case) or for real-time traffic in general.

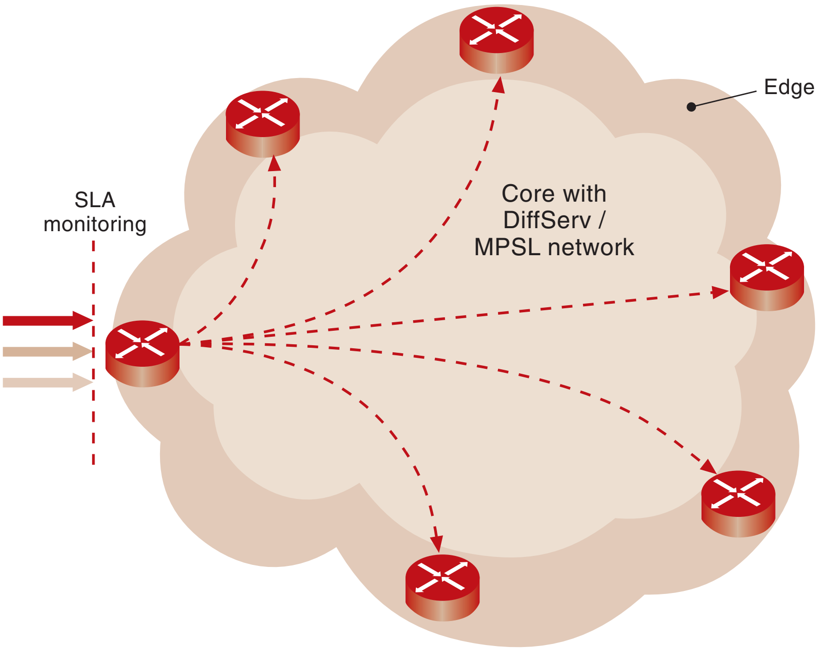
Rate reductions can be less frequent. This can be achieved by using a larger integration time for such decisions. Another solution could be to take into account past observations and for example use exponential weighting of the observations. If this is not done, only one measurement with CDR less than the reduction threshold should in any case normally not trigger a rate reduction.

At times with low traffic, no rate reductions are necessary. The control system should then keep track of bandwidth demands for the different LSPs, so that when the network starts to get loaded bandwidth can be freed from LSPs not needing it.

A problem arises when a request for bandwidth increase is rejected. Different options then apply, like (in the relevant order)

- Release bandwidth from other LSPs not needing it;
- Pre-empt lower priority traffic if possible;
- Set-up a new path using other physical resources with spare bandwidth;
- Try to re-optimise the network.

Figure 5 Overall scenario



A Control Framework for a Core Network by Use of MPLS and DiffServ

Discussion of Different Alternatives

To be able to offer QoS to users it is necessary, as we have seen, to have some control over the use of network resources. A basic part of such control builds on agreements with the users of the network (SLA). Another part could be monitoring of LSPs.

Figure 5 shows the overall scenario. User traffic is monitored and enforced at the ingress to the network (SLA monitoring). This can be in an edge router, or preferably as near to the user as possible.

In principle we can distinguish between three types of services:

i Service with connectivity to a predefined set of destination points. An example can be Virtual Leased Line service. In this case SLA specifies the allowed traffic towards these destination points (pipe model, Committed Information Rate – CIR-SLA), we have no spatial gambling and the necessary resources can be reserved in the network.

ii Service with call admission control functionality. VoIP may be implemented in this way. The call admission control will decide whether to permit or deny a given call request based on knowledge about the available resources in the network. If the conclusion is negative the service is blocked, otherwise the necessary resources can be reserved and the call set up.

iii A one-to-any service without call admission control functionality. In this case the SLA only controls the volume of the traffic flowing over the user-network interface (of each class and both ways). This is called a hose SLA (Committed Access Rate – CAR-SLA). The SLA is therefore not enough to control the volume of traffic in a given direction, i.e. we have a kind of spatial gambling on traffic volume. (In the downstream direction the SLA will be any-to-one, also called a funnel SLA.)

The latter case is of most concern from a QoS/control viewpoint. This is the service type that is most in the spirit of DiffServ and at the same time the service type that gives the least control of the traffic distribution.

In the Figure 5 scenario we have in principle three different ways of transporting traffic between edge routers. A given service component traffic will be mapped to a CoS. The traffic can then

a Be mapped to a PSC and carried by pure DiffServ;

b Be mapped to an L-LSP designated for this CoS solely;

c Be mapped to an E-LSP designated to this CoS together with other CoS (up to 8 BAs).

In either case the possibility of reserving resources must be discussed. This is of vital importance for controlling traffic and assuring QoS in a network with dynamically changing traffic.

For the time being MPLS distinguishes itself as the most promising way for introducing a control framework, both for service types ii and iii. For service type iii a measurement based solution is proposed. For service type ii the traffic volume in a given direction can be controlled by the call admission control. But our framework for dynamically changing the LSP bandwidth parameters (see above) still applies. The decision will then be based on the number of active calls instead of traffic volume measurements.

Dimensioning and re-dimensioning for service type ii can be done using

a An effective bandwidth approach for estimating the relationship between number of calls and trunk (LSP) bandwidth demand;

b A call level module for dimensioning the trunk bandwidth using traffic forecasts (call level) and a call blocking objective.

Dimensioning and re-dimensioning for service type iii requires new methods, and in the following sections we discuss a framework where network resources are re-dimensioned or re-configured based on actual measurements of traffic in the network.

A Measurement-based Approach for Traffic Control

As discussed above the possibilities of reserving resources is of vital importance for controlling traffic and assuring QoS in a network with dynamically changing traffic.

For LSPs some options can be discussed:

i Scheduling is done on the individual LSP level such that each LSP is given the reserved bandwidth. The scheduler can then be used as a shaper if no excess bandwidth is made available to the LSPs. This may not be a scalable solution with a full mesh LSP network between edge routers in the domain.

ii Scheduling is done on an aggregated level with one queue for each CoS, or rather a set of CoS. This solution is better from a scalability viewpoint, but only the aggregated bandwidth of all traffic of the same queue is guaranteed. The LSP bandwidth must therefore be enforced, or at least in some way controlled, before queueing on the output interface module (LSP parameter control).

We presume that option ii is chosen. This gives the following distribution of QoS mechanisms in an edge router with MPLS:

- Access interface upstream (direction from customer): Classification, metering, action (drop or (re)mark), forwarding;

- Access interface downstream (direction towards customer): Classification, metering, action, queueing, algorithmic dropping and possibly shaping;

- Core interface upstream: Label encapsulation, parameter control per LSP (classification, metering, action), queueing, algorithmic dropping and possibly shaping;

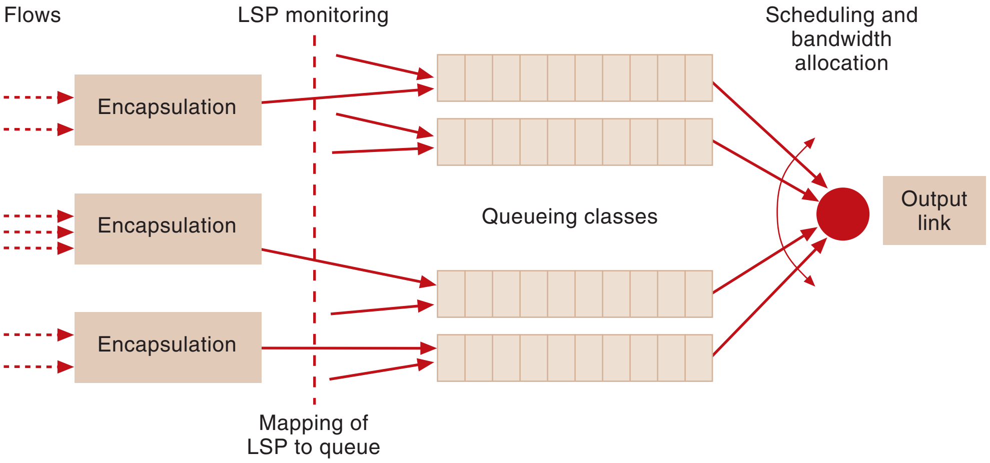
- Core interface downstream: LSP termination and forwarding.

The SLA monitoring is here indicated to take place at the access interface of the edge router. As mentioned above this should be done as near to the user as possible. The SLA monitoring could therefore be done in a router between the user and the edge router.

A conceptual model of the interface to the core network is given in Figure 6, presuming a unique mapping from LSP to DiffServ queue (e.g. L-LSPs).

The LSP monitoring in the edge routers will be the basis for the bandwidth reservations as described above. If this monitoring can be a basis for reliable control of resources, e.g. by introducing enough slack, LSP policing is not necessary.

Figure 6 Conceptual model of the edge router interface to the core network



The core router is simpler:

- Input interface: LSP label switching, possible with hierarchy functionality (push/pop including possible split);
- Output interface: LSP merge (as a consequence of label switching at the input), monitoring per LSP, CoS or per queue, queueing and scheduling.

LSP monitoring in the core router is probably not necessary, since the bandwidth reservation for a merged LSP in principle could be based on the bandwidth reservation for each individual LSP. A conceptual model of the output interface is given in Figure 7, presuming again a unique mapping from LSP to DiffServ queue.

The reservation of resources must in some way relate to the traffic actually flowing in the network and should therefore be dynamically changed based on monitoring data as described above.

The control framework can now be summarised as follows:

i User traffic is monitored at the router nearest to the user that is controlled by the operator and can do parameter control. In what follows we assume for simplicity that this is the ingress edge node.

ii The user traffic is mapped to an appropriate CoS. This can be done as part of i. Based on CoS and destination edge node the traffic is mapped to an appropriate LSP. This is assumed done in the ingress edge node, either as part of i at the access interface or at the core interface of the ingress edge node.

iii LSPs are set up between pairs of edge nodes for carrying user traffic. In core nodes LSPs carrying the same CoS (L-LSPs) and towards the same edge node, may be merged to make the configuration more scalable.

iv The LSPs are set up with reserved bandwidth using bandwidth parameters negotiated either using RSVP or LDP. The reserved bandwidth has implications for configuration of schedulers in the routers. The bandwidth can be reconfigured based on measurements.

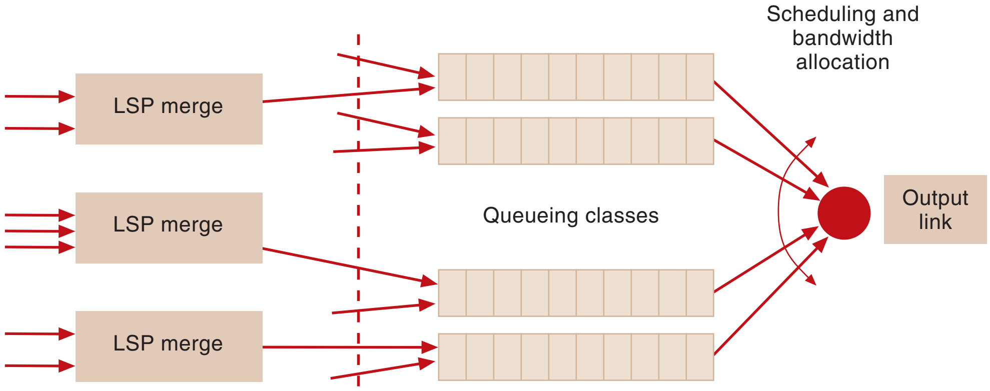


Figure 7 Conceptual model of the output interface of a core router

v The LSP bandwidth parameters are policed at the LSP ingress in the edge node. Actions to be taken (drop or re-colouring) are part of the configuration of the nodes.

vi In an operational network the LSP parameters must be dynamically updated based on traffic measurements. This is at least necessary for traffic for which bandwidth is not reserved end-to-end. Thresholds must be defined for such updating.

Conclusions

Lack of traffic control in IP networks makes it very difficult, if at all possible, to offer differentiated services with some kind of performance guarantee. A control framework has been proposed. This framework is based on the use of the Differentiated Services architecture, Multi Protocol Label Switching, monitoring of Label Switched Path traffic volume and dynamically updating of bandwidth reservations based on actual traffic.

The deployment of the proposed solution depends however on cost related to implementing monitoring hardware in the edge routers and possible technology constraints related to monitoring at high speed. Also it remains to be seen what service differentiations are possible to achieve in this type of network.

The framework should be extended to treat interdomain aspects and the use of e.g. IntServ in the access part of the network. Also related aspects like

- Use of load sharing;
- Criteria for creation and termination of LSPs;
- Criteria for triggering a re-configuration of the MPLS network, involving routing;
- Use of back-up paths and pre-emption

should be included in the framework.

References

[diffserv-new-terms] IETF. 2001. *New Terminology for Diffserv*. (draft-ietf-diffserv-new-terms-04.txt)

[diffserv-pdb-def] IETF. 2001. *Definition of Differentiated Services Per Domain Behaviours and Rules for their Specification*. (RFC 3086)

[IETF-TE] IETF. 2001. *A Framework for Internet Traffic Engineering*. (draft-ietf-tewg-framework-05.txt)

[Johnsen1999] Johnson, V. 1999. *Technology Backgrounder – Quality of Service – Glossary of Terms*. [online] – URL: www.stardust.com.

[mpls-diff-ext] IETF. 2001. *MPLS support for Differentiated Services*. (draft-ietf-mpls-diff-ext-09.txt)

[RFC1633] IETF. 1994. *Integrated Services in the Internet Architecture: an Overview*. (RFC 1633)

[RFC2474] IETF. 1998. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. (RFC 2474)

[RFC2475] IETF. 1998. *An Architecture for Differentiated Services*. (RFC 2475)

[RFC2597] IETF. 1999. *A Single Rate Three Color Marker*. (RFC 2697)

[RFC2598] IETF. 1999. *An Expedited Forwarding PHB*. (RFC 2598)

[RFC3031] IETF. 2001. *Multiprotocol Label Switching Architecture*. (RFC 3031)

The Design of Optimal Multi-Service MPLS Networks

ÅKE ARVIDSSON AND ANTHONY KRZESINSKI

Multiprotocol label switching (MPLS) extends the IP destination-based routing protocols to provide new and scalable routing capabilities in connectionless networks using relatively simple packet forwarding mechanisms.

MPLS networks carry traffic aggregates on virtual connections called label switched paths (LSPs). The first part of this paper examines under what circumstances it is advantageous to design dedicated LSPs for individual origin-destination pairs and service classes. We show that separate LSPs in most realistic cases are likely to be the preferred mode of operation.

We next consider path selection and bandwidth allocation in multi-service MPLS networks in order to optimise the overall network quality of service. The optimisation is based upon the constrained optimisation of a non-linear objective function. We present a model of an MPLS network and a computationally efficient algorithm called XFG to find and capacitate optimal LSPs. The algorithm is based on a bandwidth market where bandwidth prices determine the allocation of bandwidth to LSPs. The XFG algorithm is applied to compute optimal LSPs for a 55 node network model carrying 6 service classes.

The results above are limited to service classes typically supported by UDP, e.g. conversational voice and streaming video, where the notation of equivalent bandwidth can be applied. This is, however, not the case for service classes typically supported by TCP, e.g. interactive or background data, because of the responsiveness of the protocol. We therefore extend our work to incorporate these types of traffic and apply the XFG algorithm to compute optimal LSPs for a network of 8 nodes and 2 service classes.

Finally we use core networks of third generation cellular mobile systems as an example to show how the method can be generalised to any multi-service network and we also discuss how to include virtual private networks.

Åke Arvidsson (43) obtained the MSc and PhD from the Lund Institute of Technology, Sweden. Between 1990 and 1992 he worked at Bond University and the University of Adelaide in Australia. From 1993 to 1995 he was with the Lund Institute of Technology and in 1995 he joined the Blekinge Institute of Technology, Sweden, as Professor of Teletraffic Theory. Since 1998 he is on leave and works for Ericsson Core Network Development as Technical Expert on Data Traffic Theory. His professional interests include traffic modelling, performance evaluation, network architecture and load control.

Ake.Arvidsson@uab.ericsson.se

Anthony Krzesinski (55) obtained the MSc from the University of Cape Town and the PhD from Cambridge University, UK. In 1972 he joined the Shell Research Laboratory in Amsterdam where he worked on the development of mathematical models to predict the performance of computer systems. In 1975 he joined the Department of Computer Science at University of Stellenbosch. He is presently a Professor of Computer Science at the University of Stellenbosch. His research interests centre on the performance evaluation of communication networks.

aek1@cs.sun.ac.za

1 Background

Two major trends have dominated the telecommunications industry over the past decade, viz. wireless cellular systems such as GSM and packet switched services over IP. The two technologies have largely evolved independently of each other and the services are basically separated in the networks.

This split approach is however set to change for technological and commercial reasons. On the technology side, high speed packet switching is an integral part of third generation cellular systems, e.g. UMTS, and packet switching is becoming deployed in fixed access networks, e.g. CATV. On the commercial side, competition eats into the margins of traditional wireline services while the growth of packet switched services exposes the costs associated with separate networks.

There is thus a growing need for an integrated services network. This is by no means a new idea, in fact it has been a vision at least since the seventies, although the technologies have differed: ISDN over STM in the seventies, B-ISDN over ATM in the eighties and everything over IP in the nineties. The visions have however remained visions, albeit for different reasons. ISDN/STM was made obsolete by the development of the computer industry, B-ISDN/ATM was seen as too costly and too complicated and all-IP was questioned in relation to quality of service.

2 MPLS

The favourite candidate for service integration is MPLS [1]. The fundamental idea behind MPLS can be characterised as enhancing IP with some quality of service concepts from ATM. From this point of view, a primary feature of MPLS is the ability to perform *traffic engineering* and a secondary feature is *quality of service control* [2].

IP networks typically use OSPF or a similar protocol to find shortest routes between points. The fact that there is only one such route from any node to any other node may lead to situations where links on shortest routes are congested while other links remain idle. Traffic engineering in MPLS essentially means that traffic flows can be controlled in order to balance link loads.

Moreover, classical IP networks typically support only one service class, viz. best effort. Although proposals such as IntServ and DiffServ have been around for some time, they have not yet gained widespread acceptance, let alone deployment. IntServ suffers from scalability problems which makes it unsuitable for backbone networks, while the ability of DiffServ to provide quality of service is doubted. Quality of service control in MPLS essentially means that bandwidth can be reserved for traffic flows.

MPLS networks are accessed through ELRs and contain LSRs internally. Packets arriving at ELRs are inspected with respect to destination and classified into flows. The classification may be extended to include other attributes such as source, application class, etc. Flows are associated with unique flow labels which are used by LSRs instead of IP addresses to switch/route packets through the network. All packets of a flow may thus follow the same path (a so-called label switched path), but packets can also be classified so that packets relating to different sources or applications may follow different paths even if the destinations are identical.

As can be seen there are several similarities between LSPs and other well-known concepts, e.g. VPs or VCs in ATM. The flow labels correspond to VPIs or VCIs and the LSRs correspond to ATM cross connects or ATM backbone switches. On the other hand, VPs and VCs in ATM are established through the management system or by signalling, while LSPs are established by the IP-based LDP [3], e.g. CR-LDP [4] or RSVP-TE [5]. See also [6].

3 Multi-Service Networks

Different types of traffic have different requirements in terms of bandwidth consumption and sensitivity to delay or loss of information. To provide sufficient quality of service either the amount of transmission resources must be set to ensure that the most stringent requirements are satisfied for all flows, or a discrimination mechanism must be used that allows each flow to obtain its required quality of service.

The former approach is simple but is usually regarded as economically viable only if the traffic type with the most stringent requirements is the dominant one in terms of traffic volume. The major traffic types today are real time voice and best effort data. Voice has high requirements on delay whereas data has high requirements on loss. Although voice is still dominant in many networks, data is growing fast and is gradually becoming dominant in most networks. Moreover, it is expected that in the not so distant future video retrievals, the requirements of which are typically between voice and data, will make up a large part of the traffic in a network. The conclusion is that it does not seem realistic to satisfy the most stringent requirements for all traffic types, but some kind of differentiated quality is required.

Clearly, discrimination mechanisms only work within the limits given by the total capacity of a system, but they cannot resolve problems related to overload. To prevent the latter, overload protection in the form of connection admission control and packet policing is usually applied at the edge of a network. To determine the rules according to which these mechanisms should operate, the impact of various loads is studied by mathematical traffic models. The results are often presented as a safe region of operation, and the mechanisms are tuned to ensure that the load stays within this region.

The classical view of multi-service networks is the *integrated* one where all service classes, origins and destinations are multiplexed together directly on the physical network. The difficulty with this approach is that the traffic models are quite complex, and that combined models of all traffic types are even more complex because of the additional difficulties in combining the different kinds of models used for different service classes. Considerable simplifications must thus be made to obtain a tractable result. The consequences are that the accuracy of a safe region is questionable and additional margins must be added, and that there is no direct link between modelling errors and performance problems since problems for one service class may be related to modelling errors for any class or for a combination of service classes.

A contrasting view is the *separated* one where all service classes, origins and destinations are supported by dedicated logical end-to-end links, such as LSPs, which identify routes and reserve bandwidth. In this approach transmission resources are partitioned between various service classes and node pairs. This means that admission controls operate on dedicated resources based on single class traffic models and there is no need for joint models of all classes. Consequently, modelling errors for one class will not impact other classes and a failure in meeting a performance target for a certain class is corrected by modifying the particular model in question. Moreover, new classes can be added to the network without reworking the complete model of all classes and without impairing the performance of other service classes.

It may be argued that this way of partitioning will be less efficient in exploiting statistical multiplexing gain. However,

- *slow* variations may be handled just as efficiently by redistributing the resources between the logical links [7, 8, 9, 10]; for

- *fast* variations and service classes with *different* time scale characteristics, it has long been known that there is little or no gain to be had from multiplexing [11]; and for

- *fast* variations and service classes with *similar* time scale characteristics, it has been suggested that the gain obtained from multiplexing tends to be outweighed by increasing overhead costs [12].

- Moreover, as indicated above, a partitioned scheme permits more aggressive multiplexing than an integrated one since the former avoids the compromises of a joint traffic model, but permits the use of more accurate single class traffic models.

The reason behind the first point is that the set of LSPs may be re-designed as required. The proposal in [7] requires that ELRs monitor offered traffic during intervals of t_M time units, with new intervals commencing every t_Uth time unit. Traffic estimates are forwarded to an NMC which computes updated LSPs and analyses the result. If implementing the new design appears profitable, the necessary information is sent back to the ELRs and the design is implemented by an LDP. Transmitting traffic information to the NMC, computing and analysing the design, returning results to the nodes, and implementing the design is assumed to take a total of t_E time units. There is a trade-off between the resources spent on management actions such as altering LSPs and the associated increase in carried traffic. To compare the two and optimise the strategy, it is proposed in [7] to associate a profit for carried traffic and a cost C_T for each updating attempt (transmission of data to the NMC, computing and analysing a design), and a cost C_I for implementing a new design. Other cost models and similar proposals which are independent of the NMC are discussed in [8, 9] and in several papers in [10].

The second point simply says that the statistics on which multiplexing relies must apply to all classes. This means, for example, that it must be possible and meaningful to buffer one class during the busy periods of another class. The early work reported in [11] showed that the multiplexing gains obtained from mixing voice and data are limited to quality of service improvements but do not impact decisions regarding engineering. The work considered the SENET concept where bandwidth is divided in time between voice and data. The boundary between voice segments and data segments could either be movable or fixed depending on whether bandwidth reserved for, but not used by, voice was made available to data or not. The work in [11] showed that the gains obtained from the movable boundary were limited by the fact that the dynamics of voice (connection holding times) are very slow compared to the dynamics of data (packet inter-arrival times). The fact that the bandwidth for data is controlled by voice means that data will experience good service when few connections are in progress but poor service when many connections are in progress. In fact, the work showed that during the latter intervals data traffic will be congested to the extent that the service appears useless. This means that, because of the different time scales, there is no statistical gain to be obtained from multiplexing the two services.

The third point refers to the fact that full multiplexing requires link-by-link processing whereas only end node processing is required for LSPs. Depending on the relative costs of transmission and processing, more powerful links to support LSPs may be cheaper than more powerful nodes to support full multiplexing. As a simple example [12], consider two flows f_1 and f_2 of the same service type with traffic demands ρ_1 and ρ_2 respectively. Let f_1 traverse a route from o_1 to d_1 and f_2 from o_2 to d_2 which both span h hops. Assume that the two routes have one physical link ℓ in common and that it is the bottleneck link in the sense that its bandwidth C_ℓ determines the grade of service offered to the two flows. Without LSPs, all of the bandwidth B_ℓ of ℓ is available to f_1 and f_2 whereas, with LSPs, f_1 has access to a capacity C_1 and f_2 to a capacity C_2, with $C_1 + C_2 = B_\ell$. Moreover, without LSPs, it takes h rounds of processing and control signalling to perform CAC (one per physical link) and each packet must be analysed h times whereas, with LSPs, it takes one hop of processing and control signalling to perform CAC (on the logical link) and each packet must be analysed once. In addition, a "once-and-for-all" cost of $h = 5$ hops of processing and management signalling is required to establish an LSP the cost of which is depreciated over the expected life time of the LSP $T = 10$ connection holding times. The advantage of not having LSPs is a higher degree of statistical multiplexing whereas the advantage of LSPs is less processing and control signalling. Figure 1 shows the overhead cost relative to traffic gain at which the two advantages even out for the case where the loss without LSPs is fixed to 1 %. It is seen that, e.g. when f_1 and f_2 amount to 100 erlangs LSPs will be preferred if the cost of signalling and processing per connection exceeds 0.1 % of the revenue per connection. More elaborate examples in [12], which consider a set of realistic networks with a multitude of flows of different magnitudes, show similar results.

The fact that the arguments are in favour of separation does not mean that all service classes should necessarily have resources of their own. On the contrary, service classes with similar statistical characteristics and similar quality of service demands may very well share resources. A simple strategy may be to begin with completely separated networks and later, when enough operational experience is gained, merge some classes. The resources that are released at this stage can be used to satisfy a growing demand.

The conclusion is that separation will, in many cases, be the preferred option. The evolution of MPLS and current interest in the concept confirm this assumption. A highly relevant question is then how to design and manage separated networks of LSPs on top of a common MPLS infrastructure.

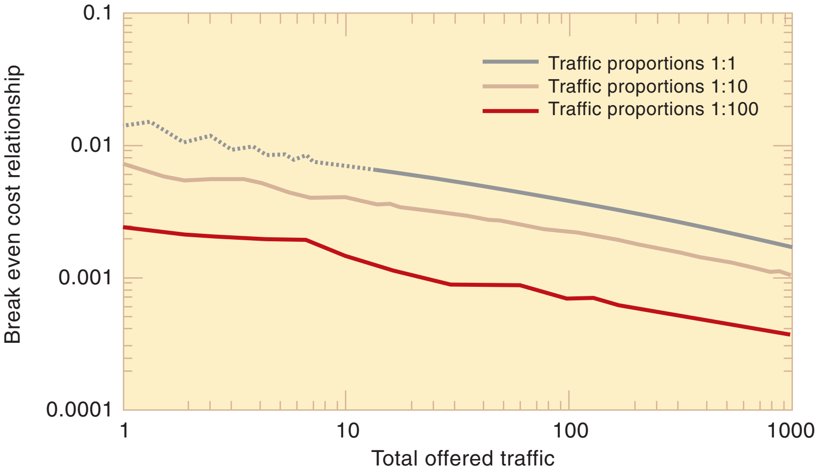


Figure 1 Comparison of sharing and partitioning of bandwidth on a link

4 The Design of Optimal MPLS Networks

We now turn to the problem of how to design a set of LSPs such that each origin-destination (O-D) pair is connected by one or more LSPs per service class. A survey of algorithms which are more or less suitable for this purpose can be found in [13].

We present two MPLS models. The first model, presented in this section, finds and capacitates optimal LSPs which support connection oriented aggregates using, e.g., UDP for which the notation of equivalent bandwidth applies and some CAC mechanism is in place. The second model, presented in Section 5, finds and capacitates optimal LSPs which support connectionless aggregates using TCP for which equivalent bandwidth makes no sense and best effort applies.

4.1 The LSP Design Problem

Consider a network which consists of N nodes and L physical links. The network supports A aggregates, each of which is characterised by its origin ELR, destination ELR and service class. Each aggregate a corresponds to a unique triple (o,d,s): let $n(a) = (o,d)$ denote that the origin and destination nodes of aggregate a are o and d respectively and let $s(a) = s$ denote that the service class of aggregate a is s.

Let λ_a denote the Poisson arrival rate of connection requests in aggregate a. Let $1/\mu_a$ denote the mean connection holding time in aggregate a. The holding time distribution is general. Let $\rho_a = \lambda_a / \mu_a$ denote the load offered by aggregate a. A connection in aggregate a generates revenue at a rate θ_a per time unit.

Let B_ℓ denote the bandwidth of the link ℓ between $n(\ell) = (o,d)$ such that $B_\ell > 0$ denotes the existence of a physical link from node o to node d.

A route r is a non-cycling sequence of physical links connecting an origin node to a destination node. An LSP on a single link route where $|r| = 1$ is said to follow a *direct* route and an LSP on a multi-link route r where $|r| > 1$ is said to follow a *transit* route.

Let $\mathscr{R}_a$ denote the set of LSPs including the direct one (if it exists) that is used to carry aggregate a. Let $\mathscr{L}_\ell$ denote the set of LSPs including the direct LSP that uses link ℓ.

Let x_r denote the bandwidth assigned to LSP r. Let $\boldsymbol{x}_a = (x_r)_{r \in R_a}$ denote the bandwidths assigned to the LSPs in $\mathscr{R}_a$. The capacity $C(a,\boldsymbol{x}_a)$ of the LSP set $\mathscr{R}_a$ used by aggregate a is given by

$$C(a, x_a) = \sum_{r \in \Re_a} f_{s(a)}(x_r) \qquad (1)$$

where $f_{s(a)}(x)$ is the equivalent number of service class $s(a)$ circuits configured on LSP r which has bandwidth x. The equivalent number of circuits is the maximum number of simultaneous connections that can be supported. The function will typically be non-linear unless peak rate allocation is applied.

Let $E(C(a,\boldsymbol{x}_a),\rho_a)$ denote the blocking probability (objective function) experienced by aggregate a connections with load ρ_a on an LSP set of capacity $C(a,\boldsymbol{x}_a)$. The rate $F(a,\boldsymbol{x}_a)$ at which aggregate a connections generate revenue when the LSP configuration is $\boldsymbol{x}_a$ is

$$F(a,\boldsymbol{x}_a) = \theta_a \rho_a (1 - E(C(a,\boldsymbol{x}_a), \rho_a)). \qquad (2)$$

The *LSP design problem* is specified in terms of the following constrained non-linear optimisation problem: Find the LSP configuration $\boldsymbol{x}_{\text{opt}}$ that maximises the revenue earning rate $F(x)$

$$F(x_{\text{opt}}) = \max_x F(x) = \max_x \sum_{a=1}^{A} F(a, x_a)$$

where $\boldsymbol{x} = (\boldsymbol{x}^1, ..., \boldsymbol{x}^A)$ is subject to the constraints

$$x_r > 0$$

for all routes r, and for all links ℓ

$$\sum_{r \in A_1} x_r = B_\ell$$

In words, the constraints require that each LSP has a strictly positive bandwidth and that the bandwidths of the LSPs passing through link ℓ in total use the entire bandwidth of link ℓ.

The necessary condition for $\boldsymbol{x}$ to be a local optimum for $F(\boldsymbol{x})$ is that for any route r in the LSP set $\mathscr{R}_a$

$$\frac{\partial}{\partial x_r} F(a, x_r) = \sum_{\ell \in r} \frac{\partial}{\partial x_\ell} F(\cdot, x_\ell). \tag{3}$$

In words, equation (3) says that the change in revenue obtained by moving an infinitesimal amount of bandwidth to route r of aggregate a is equal to the revenue lost in acquiring this bandwidth from aggregates whose LSP sets include direct LSPs over the links of r, and vice versa.

4.2 XFG: An LSP Optimisation Algorithm

We wish to design an MPLS network were each aggregate is supported by one or more LSPs. If a flow is routed over more than one LSP, packets belonging to the same flow may arrive out of order at the destination. This can be avoided by a systematic sub-classification of packets such that any flow in an aggregate will always use the same LSP. For example, if two LSPs with equal bandwidth support an aggregate, packets may be split between the LSPs according to the parity of the full destination address.

4.2.1 A Bandwidth Market

The XFG algorithm is based on the concept of bandwidth cost. Each aggregate computes the price that it is willing to pay in order to buy more bandwidth, and the price at which it is willing to sell off bandwidth. An under-capacitated aggregate will buy more bandwidth on an existing LSP or establish a new LSP according to where the largest revenue increase can be made at the smallest bandwidth cost. An over-capacitated aggregate will sell off bandwidth on the LSP where the largest bandwidth cost can be obtained for the smallest revenue decrease.

Bandwidth is traded in units the size of which is typically determined by scheduling mechanism constraints in the LSRs and quality of service requirements from users. For example, LSRs that schedule complete IP packets with a maximum size of 1540 bytes over links the bandwidth of which is 2.048 Mbps will be able to serve at most 2.048×10^6 / (1540 × 8) = 170 packets per second. If the quality of service requires that a reserved bandwidth is to be realised on a 100 ms time scale a link will correspond to 0.1 × 170 = 17 units. Alternatively, if the LSRs schedule ATM cells, at most 2.048×10^6 / (53 × 8) = 4830 cells per second can be served, hence a link corresponds to 483 units under the same resolution requirements.

4.2.2 Link and LSP Prices

The gain $Q_a(c)$ obtained by allocating u additional units of bandwidth to an LSP supporting aggregate a is given by the increase in revenue when the bandwidth of the LSP is increased from c to $c + u$. Thus

$$Q_a(c) = F(a,c+u) - F(a,c) \tag{4}$$

where the link revenue function $F(a,c)$ is given in equation (2).

When the bandwidth of the aggregate a LSP from node o to node d is increased from c to $c + u$, the additional u units of bandwidth are obtained from aggregates with direct LSPs on all links ℓ on the route. The cost $q_{a'(\ell)}(c)$ of acquiring u units of bandwidth from an aggregate $a'(\ell), a'\ell \in \mathscr{R}_{a'}$, is given by

$$q_{a'}(c) = F(a',c) - F(a',c-u) \tag{5}$$

where for each link ℓ, class s($a'(\ell)$) is the cheapest provider of bandwidth on link ℓ and c is the class s($a'(\ell)$) bandwidth of link ℓ.

Equations (4) and (5) approximate the derivatives of equation (3). A cubic spline is fitted to the values computed for equations (4) and (5) to ensure that the values of left- and right derivatives are identical.

The total cost $q_r(x)$ of acquiring a unit of bandwidth from each link along the route $r = (\ell_1, ..., \ell_J)$ is

$$q_r(x) = \sum_{j=1}^{J} q_{a'(\ell_j)}(c)$$

4.2.3 The Algorithm

The XFG algorithm uses the model of a bandwidth market to solve the LSP design problem by allocating capacity to LSPs in a series of *transactions*. The algorithm executes in a loop and each iteration of the loop implements one transaction. A transaction can either be a multi-link LSP buying one unit of bandwidth from a set of single link LSPs or a set of single link LSPs buying one unit of bandwidth from a multi-link LSP. The transaction chosen is the one that yields the highest profit, except that the inverse of the previous transaction is barred.

The profit of a transaction is defined as the difference between a buying price and a selling price. Each LSP and each link compute the price at which it is willing to buy or sell bandwidth. A search is then made over all aggregates to determine the profit from (i) buying, and (ii) selling bandwidth on any of the existing LSPs, and (iii) buying bandwidth on a new LSP. The sellers in the latter case are the links along the current least cost path which is computed from current link prices by Floyd's shortest path algorithm.

After each transaction the prices are re-evaluated and the algorithm proceeds to the next iteration. The transactions continue until no profit can be made from buying/selling bandwidth and the algorithm halts.

An overview of the operation of the XFG algorithm is presented below, see [14, 15] for a complete description.

1 **Initialisation**. Establish LSPs between all nodes which are interconnected by direct links and assign all bandwidth on the links to these LSPs. Compute the prices they would charge for selling one unit of bandwidth.

2 **Compute shortest paths**. Let the price of bandwidth denote the cost of a link and compute the least cost paths for all O-D pairs.

3 **Find the most attractive allocation**. Examine all profits, i.e. buying prices less selling prices, that would result if multi-link LSPs bought more bandwidth from single link LSPs or if new multi-link LSPs were created by buying bandwidth from single link LSPs on the least cost paths. Identify the transaction that offers the maximum profit.

4 **Find the most attractive de-allocation**. Examine all profits, i.e. selling price less buying price, that would result if multi-link LSPs sold bandwidth to single link LSPs. Identify the transaction that offers the maximum profit.

5 **Convergence test**. If the profits from the most attractive allocation and de-allocation are both negative or sufficiently close to zero, then stop.

6 **Perform the most attractive transaction**. If the profit from the best allocation exceeds that of the best de-allocation, then remove one unit of bandwidth from the single link LSPs on the path of the existing or new multi-link LSP and add one unit of bandwidth to the existing or new multi-link LSP. Update the bandwidth prices of the affected LSPs.

If the profit from the best de-allocation exceeds that of the best allocation, then add one unit of bandwidth to the single link LSPs on the path of the existing multi-link LSP and remove one unit of bandwidth from the existing multi-link LSP. Update the bandwidth prices of the affected LSPs.

7 **Loop Statement**. Go to step 2.

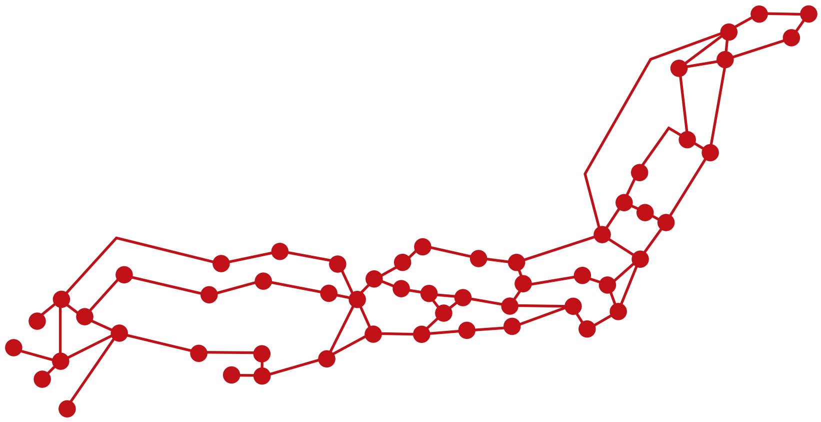
The XFG algorithm belongs to the class of so-called greedy algorithms where the action taken at each optimisation step is the one which immediately gives the highest reward without considering long term impacts. However, the final result is optimal if the bandwidth prices are convex, and if the unit of bandwidth can be made infinitely small. The former condition is fulfilled by, e.g., using the Erlang-B formula for the objective function $E(\cdot,\cdot)$ in equation (2), but the latter restriction will in practice put a limit on the optimality.

It is noted that the number of iterations required by the algorithm appears to depend on the number of bandwidth units in the network. This seems to suggest a conflict between accuracy (which requires small units) and speed (which requires large units). This is however not the case as a large unit can be used initially to quickly compute an approximate solution after which successively smaller units can be used to refine the solution to within practical limits.

4.3 An Application

Consider the network presented in Figure 2 consisting of 55 nodes connected by 71 bi-directional links. The network contains 1,485 O-D pairs and carries 6 service classes. Peak rate allocation is presumed and the effective bandwidth requirement of connections in aggregate $a = 1, \ldots, 6$ are 1, 3, 4, 6, 24 and 40 bandwidth units respectively. Note that, although not used in this example, Equation (1) also allows for variable bit rate allocation where the effective bandwidth of a connection depends upon the statistical gain

Figure 2 A 55 node network



Route length	Bandwidths		LSPs	
	C_i	C'_i	L_i	L'_i
0	284,074	0	8,591	0
1	11,885	148,839	1,191	426
2	10,753	21,781	1,099	718
3	7,527	24,859	812	1,056
4	5,050	24,212	552	1,253
5	2,824	22,244	426	1,359
6	1,806	17,586	230	1,303
7	1,023	13,472	163	1,170
8	544	10,920	79	1,082
9	321	8,946	50	998
10	78	7,192	18	877
>10	81	25,915	21	2,990
Total	325,966	325,966	13,232	13,232

Table 1 Distribution of the allocated bandwidth C_i on LSPs of normalised length i; distribution of the allocated bandwidth C'_i on LSPs of un-normalised length i; distribution of the normalised L_i and un-normalised L'_i LSP lengths

on the LSP. The transmission capacity of each physical link is 10,000 units. The 25 O-D pairs that carry low traffic are connected by one transmission link, the 37 O-D pairs that carry more traffic are connected by two links, 8 O-D pairs are connected by three links and 1 O-D pair is connected by four links. The physical links contain 1,270,000 units of transmission capacity. The traffic load offered to each aggregate a is $\rho_a = 1.0$. The revenue rates

$$\theta_a = \begin{cases} 1 & s(a) = 1, 2, 3, 4 \\ 2 & s(a) = 5 \\ 4 & s(a) = 6 \end{cases}$$

ensure that narrowband connections do not exclude broadband connections from service.

The XFG algorithm requires 7 minutes of execution on a Pentium III 1.5 GHz processor to compute the solution. The 1,270,000 units of physical transmission capacity are used to configure 13,232 LSPs. With reference to Table 1, 148,839 units of bandwidth are configured on 426 (71 × 6) single link LSPs (direct routes) and 1,121,161 units of link bandwidth are used to configure 151,212 units of bandwidth on 12,806 multi-link LSPs (transit routes).

The lengths of the LSPs and their bandwidth assignments are shown in Table 1. Let u_r denote the *un-normalised length* of an LSP r which is equal to the number of links in the route. Let $n_r = u_r - m_r$ denote the *normalised length* of an LSP r, where m_r is the number of links in the shortest LSP connecting the O-D pair. A route r with normalised length $n_r = 0$ is thus the shortest route connecting the O-D pair, and a route r with normalised length $n(r) = 1$ is thus one link longer than the shortest route connecting the O-D pair. 74 % of the LSPs are constructed on the shortest and the shortest-but-one paths connecting the O-D pairs: 91 % of the bandwidth is configured on these shortest LSPs.

5 An Objective Function for TCP

The LSP design problem presented in Section 4.2.1 – 4.2.3 and the application in Section 4.3 made use of the Erlang-B formula as an objective function. This approach is suited to service classes such as voice, for which CAC and equivalent bandwidth are applicable. However, it does not work for other service classes such as data which typically are connectionless (hence CAC does not apply) and adapt their transmission rates to the congestion encountered (hence equivalent bandwidth does not apply). The dominating protocol for such services is TCP. In this section we present an objective function which can be used to take the main features of TCP into account when finding and capacitating optimal LSPs. The work is inspired by [17] and also related to, e.g. [18, 19, 20].

5.1 A Simple Model of TCP Traffic Performance

This section presents a simple model [21] of TCP Reno [22, 23] that is restricted to (i) single path transfers, (ii) bulk data transfers for which the initial slow start phase of TCP can be neglected, and (iii) low loss systems for which time-outs can be neglected. The model is based on first order approximations and mean field theory, where all relevant parameters can be described by their averages and these averages apply to all flows in an aggregate.

The assumptions (i) – (iii) as well as the considerable modelling simplifications below clearly suggest that the numerical results obtained may be questioned. The point is, however, not to provide detailed *quantitative* results but to show how TCP traffic can be included from a *qualitative* point of view. This is an important point, because the very nature of TCP traffic is different from traditional services in terms of e.g. the best effort concept and the ability to adapt to congestion.

Work in progress includes a more elaborate TCP model which also accounts for short file transfers, slow start and time-outs. The queuing/loss model is also elaborated upon.

5.1.1 A Single Path Model

Consider a uni-directional path between two edge routers to which users and servers are connected. An infinite[1] population of users request file downloads of e.g. web pages from servers. Both users and servers have direct access to ELRs of an MPLS network. The bandwidth of the access links is limited on the user side and infinite on the server side. No losses occur on the access links. The capacity of the servers is also assumed to be infinite.

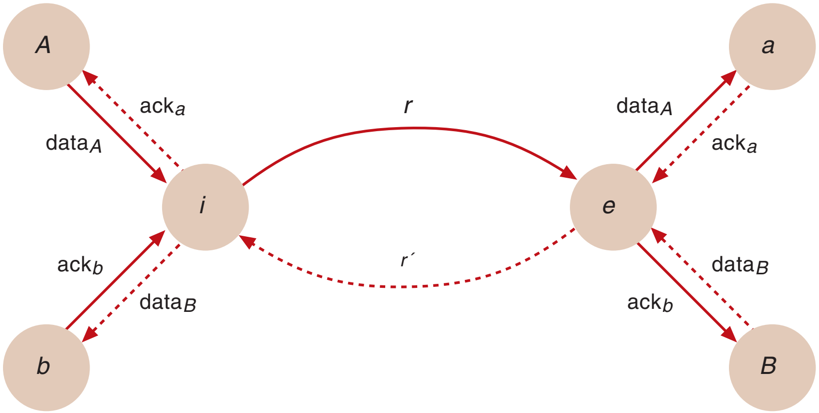


Figure 3 Packet flow in the forward and reverse directions

Consider two ELRs i and e connected by an LSP r from i to e and an LSP r' from e to i. A server A is attached to router i and a user a is attached to router e. Figure 3 illustrates the packet flow when a user a requests service from server A: data packets of average length p_{usr} are transferred from the server A to the user a via LSP r and acknowledgement packets of average length p_{ack} are returned from the user a to server A via LSP r'. The situation when a user b attached to the router i requests service from a server B attached to the router e is also illustrated in Figure 3. Each LSP r and r' may thus carry both data and acknowledgement packets.

We model an edge router transmitting packets on an LSP r by an M/M/1/K queue which is characterised by its transmission rate μ_r bits per second and the maximum number $K - 1$ of packets that it can store in its buffer memory. The packets in this model represent the weighted average of the data and acknowledgement packets – see equations (7) and (15) below. With reference to Figure 4, let γ_r denote the rate at which servers offer data packets to LSP r and let $\gamma_{r'}$ denote the rate at which servers offer data packets to LSP r'. Let $e_{r'}$ denote the loss probability for packets on path r'. Then $\gamma_{r'}(1 - e_{r'})$ denotes the rate at which users offer acknowledgement packets to LSP r to acknowledge the data packets that they have successfully received from servers.

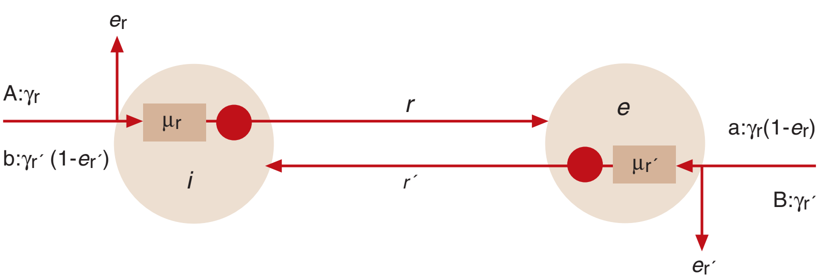


Figure 4 Model of two edge routers connected by LSPs

The performance of each LSP r will be defined by a fixed point equation in a multi-dimensional space which includes the packet loss probability e, the load ρ offered to the LSP, the round trip time t, the user window size w, the packet rate λ per TCP flow and the number s of TCP flows in progress. Since packets flow in both directions we are looking for a fixed point which consists of two sets $\{e,\rho,t,w,\lambda,s\}$: one for LSP r and one for LSP r'. The same reasoning, mutatis mutandis, applies to acknowledgements on LSP r'.

The first step in the derivation of the fixed point equation is to derive an expression for the round trip time, which is the average time taken for a data packet to be transmitted from the server to the user and an acknowledgement packet to be returned from the user to the server. We begin by observing that in terms of the M/M/1/K model the probability e that a packet will be lost is related to the offered load r as

$$e_r = \frac{\rho_r^K}{\sum_{k=0}^K \rho_r^k} = \begin{cases} \frac{1-\rho_r}{1-\rho_r^{K+1}}\rho_r^K & \rho_r \neq 1 \\ \frac{1}{K+1} & \rho_r = 1. \end{cases} \quad (6)$$

The rate γ at which data packets are offered to the path can be computed from the equation

$$\rho_r = \frac{\gamma_r p_{\text{usr}} + \gamma_{r'}(1 - e_{r'})p_{\text{ack}}}{\mu_r}. \quad (7)$$

The average time y_r to transmit a packet is then given by

$$\rho_r = \frac{\gamma_r p_{\text{usr}} + \gamma_{r'}(1 - e_{r'})p_{\text{ack}}}{\mu_r}. \quad (8)$$

The average number q of packets in the system is given by the M/M/1/K formula

$$q_r = \sum_{k=0}^{K} k \frac{\rho_r^k}{\sum_{k=0}^K \rho_r^k}$$

$$= \begin{cases} \frac{\rho_r}{1-\rho_r} \frac{1-(K+1)\rho_r^K + K\rho_r^{K+1}}{1-\rho_r^{K+1}} & \rho_r \neq 1 \\ \frac{K}{2} & \rho_r = 1. \end{cases} \quad (9)$$

[1] *Throughout this section "infinite" denotes a value which is large enough for its precise value to be irrelevant.*

The round trip time *t* of a data packet can now be calculated as

$$t_r = q_r y_r + \frac{p_{\text{usr}}}{\mu_r} + \tau + \frac{p_{\text{usr}}}{\mu_{\text{acc}}}$$
$$+ \frac{p_{\text{ack}}}{\mu_{\text{acc}}} + q_{r'} x_{r'} + \frac{p_{\text{ack}}}{\mu_{r'}} + \tau \qquad (10)$$

where τ is the propagation delay. The terms in the above equation correspond respectively to a data packet being queued at the forward LSP *r*, transmitted on the forward path, propagated to the far end, and transmitted on the access link followed by an acknowledgement packet being transmitted on the access link, queued at the backward LSP *r'*, transmitted on the backward path and propagated to the near end.

The second step is to compute the rate λ of data packets per TCP flow. We do this by first deriving an equation which relates the user window size *w* to the packet loss probability *e* by considering the evolution of the window size over time, both of which are modelled as discrete in terms of packets and round trip times respectively. Given the window size and the round trip time, we can compute the rate λ.

For a TCP Reno connection performing congestion avoidance and for which all losses are detected by triple acknowledgements such that fast recovery and fast retransmit apply, we may write

$$w_r(n+1) = (w_r(n)+1)(1-e_r)^{w_r(n)}$$
$$+ \frac{w_r(n)}{2}\left(1-(1-e_r)^{w_r(n)}\right).$$

This equation states that *w*(*n*) packets are transmitted during the n^{th} time interval: the window size will be incremented by one in the next interval if all of the transmissions are successful and halved otherwise. The probability of an increase is $(1 - e)^{w(n)}$ and the probability of a decrease $1 - (1 - e)^{w(n)}$. A steady state average may now be obtained by letting $n \to \infty$, which leads to

$$w_r = (w_r+1)(1-e_r)^{w_r} + \frac{w_r}{2}\left(1-(1-e_r)^{w_r}\right).$$

Finally, a lower limit of $w \geq 1$ and an upper limit of $w \leq w_{\text{max}}$ are applied. The lower limit models the protocol while the upper limit can be set to account for restrictions imposed by the receiver. Note, however, that the user window size *w* refers to an average, hence the bounds are not strict.

The rate λ of data packets per individual TCP flow is related to the user window size *w* and the round trip time *t* as

$$\lambda_r = \frac{w_r}{t_r + t_r e_r/(1-e_r)}. \qquad (11)$$

This equation states that one window size of data is sent during one round trip time after which any lost packet is re-transmitted until it is successful. Again a limit $\lambda_r \leq \lambda_{\text{max}}$ is applied which can be set to account for the access rate $\mu_{\text{acc}} / p_{\text{usr}}$.

The third step in the derivation of the fixed point equation is to compute an improved estimate of the load ρ_r offered to LSP *r* based on the performance according to Equation (6) – (11).

Several TCP flows may be in progress in parallel. Let υ_a denote the total file transfer request rate (requests per second) on an LSP *r* supporting aggregate *a*. The average number s_a of TCP flows in progress is related to the request rate υ_a as

$$s_a = \alpha_a \upsilon_a \frac{n_a}{\lambda_r} \qquad (12)$$

where α_a is an attraction factor which reflects the performance experienced by the users, and n_a is the total number of packets (including retransmissions) transmitted in order to satisfy a request. The equation follows from Little's result where αv is the arrival rate to a system and n/λ is the time spent in the system.

The attraction factor α is intended to model the relationship between the performance of a network and the traffic offered to it. The rationale for this is that users tend to request more web pages the smaller the presentation delay, and vice versa. The presentation delay is in turn related to queuing, transmission, propagation and loss recovery. Given the constraints imposed by user access links, it is suggested to represent presentation delay as being proportional to the throughput rate $\lambda(1 - e)$ of un-errored packets normalised by the maximum rate $\mu_{\text{acc}} / p_{\text{usr}}$ at which users can extract packets from the access link. Assuming a linear relationship between presentation delay and attraction we obtain

$$\alpha_a = \frac{\lambda_r(1-e_r)}{\mu_{\text{acc}}/p_{\text{usr}}}. \qquad (13)$$

Equation (13) implies that full attraction $\alpha = 1$ occurs when the bottleneck has been shifted from the MPLS network to the user access. The transmission factor is obtained as

$$n_a = \frac{\varphi_a}{p_{\text{usr}}} \frac{1}{1-e_r} \qquad (14)$$

where φ_a denotes the average size (in bits) of a requested file in aggregate a. The first factor in equation (14) represents the number of packets that must be transmitted and the second factor represents the number of times each packet is transmitted before it is successfully received. Inserting equations (13) and (14) into equation (12) we notice that the average number s_r of TCP flows in progress on path r is constant

$$s_a = \nu_a \varphi_a / \mu_{\text{acc}}.$$

Note, however, that the request completion time will depend on the performance of the LSP, hence the satisfied demand, i.e. the session throughput per time unit, will increase the better the performance, and vice versa.

The total rate at which packets are offered to a TCP connection is adjusted in response to the congestion encountered along the route. We therefore adjust the offered load ρ to match the user packet rate λ as

$$\rho_r = \frac{s_a \lambda_r p_{\text{usr}} + s_{a'} \lambda_{r'} (1 - e_{r'}) p_{\text{ack}}}{\mu_r}. \qquad (15)$$

Equations (6) through (15) define a system of two simultaneous non-linear equations

$$\rho_r = f_r(\rho_r, \rho_{r'})$$
$$\rho_{r'} = f_{r'}(\rho_r, \rho_{r'})$$

which can be solved numerically by applying Newton's method with numerically computed Jacobians.

The revenue function corresponding to the objective function (2) when a is a TCP aggregate is defined similarly, but with the "hard" blocking probability $E(\cdot,\cdot)$ enforced by CAC replaced by the "soft" blocking probability reflected in the attraction factor α, hence

$$F(a, \boldsymbol{x}_a) = \theta_a \nu_a \varphi_a \alpha_a \qquad (16)$$

where θ_a is the revenue per bit, $\nu_a \varphi_a$ is the number of bits offered per time unit and $\alpha_a = \alpha_a(\boldsymbol{x}_a)$ is the fraction of bits carried under configuration $(\boldsymbol{x}_a)$.

Note that in more elaborate models of TCP and possibly also in models of higher layer application protocols, the blocking factor α may include "hard" events related to protocols. In TCP e.g., a retransmission time-out failure will occur when repeated losses and time-outs have forced the value of the time-out to its maximum value 64 seconds.

5.1.2 Extension to Multi-Path Routing

The model can readily be extended to the case where several LSPs are used for each TCP aggregate. Let there be K such LSPs and let superscript (k) refer to the k^{th} LSP. Let the fraction of the traffic carried on LSP k be denoted by $\phi^{(k)}$ and assume that traffic is balanced to yield equal loads on all paths

$$\phi^{(k)} = \frac{\lambda^{(k)}}{\sum_{k=1}^{K} \lambda^{(k)}} = \frac{\mu^{(k)}}{\sum_{k=1}^{K} \mu^{(k)}}.$$

Equal loads (and equal buffer memories) result in equal packet loss probabilities e and equal queues q for all paths. An average user is assumed to see an average path which behaves like a weighted sum of the individual paths.

Re-writing equation (8) for the average packet transmission time results in

$$y_r^{(k)} = \frac{\gamma_r p_{\text{usr}} + \gamma_{r'} (1 - e_{r'}) p_{\text{ack}}}{\gamma_r + \gamma_{r'} (1 - e_{r'})} \frac{1}{\mu_r^{(k)}} \qquad (17)$$

since the scaling factors $\phi^{(k)}$ which apply to γ cancel out. Rewriting equation (17) as $y_r^{(k)} = A_r / \mu_r^{(k)}$ and letting $\mu_r = \Sigma_{k=1}^{K} \mu_r^{(k)}$, the average packet transmission time x is

$$y_r = \sum_{k=1}^{K} \alpha^{(k)} \frac{A_r}{\mu_r^{(k)}} = \sum_{k=1}^{K} \frac{\mu_r^{(k)}}{\mu_r} \frac{A_r}{\mu_r^{(k)}}$$
$$= \sum_{k=1}^{K} \frac{A_r}{\mu_r} = \frac{K A_r}{\mu_r} \qquad (18)$$

which implies that the net effect of distributing transmission capacity between K equally loaded paths is that the transmission time is scaled by K.

Finally, the average window sizes w, user packet rates λ, and number of transfers in progress s are obtained for multiple paths as above.

5.2 An Application

Consider the network [16] presented in Figure 5, which is a fictitious representation of the core NSF ATM backbone consisting of 8 nodes connected by 20 uni-directional links. The transmission capacity of each link is 5,624 units (1 unit equals 64 kbits/sec). The double lines between nodes 3 and 4 and nodes 7 and 8 indicate that two uni-directional links connect these nodes.

The network carries two TCP services which are aimed at domestic and business users respectively. The two categories differ in terms of the speed of their access links and the sizes of their requests. For domestic users we have $\mu_{\text{acc}} = 48$ kbps and $\varphi = 30$ kbytes, while for business users

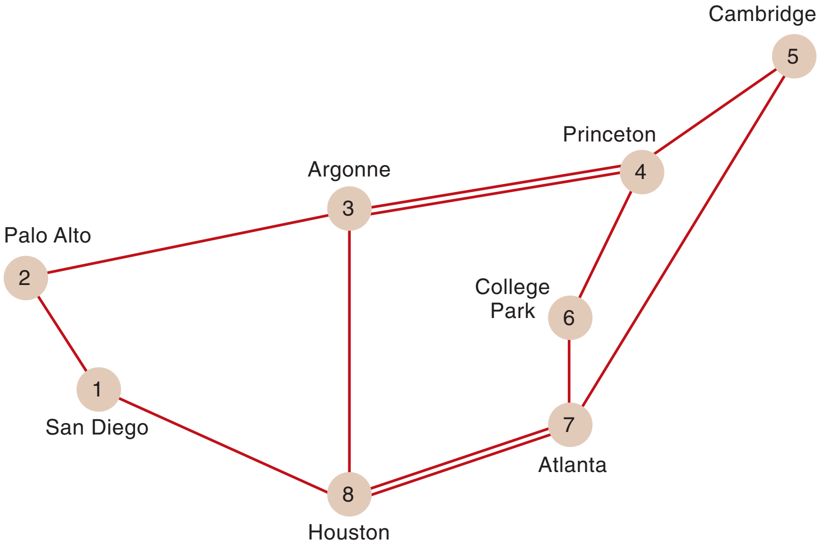


Figure 5 The core NSF network

we have μ_{acc} = 2,048 kbps and φ = 60 kbytes. The data and acknowledgement packets are of size 1,500 and 40 bytes respectively. The link propagation delay τ, which includes nodal processing delays, is for reasons of simplicity fixed to 100 ms and the maximum window size w_{max} = 64. The edge routers have (logically) separate buffers for each aggregate. All buffers are of size 5 packets and the revenue θ earned per bit carried is 1 and 3 for domestic and business users respectively. The request arrival rates differ between O-D pairs as shown in Table 2, but not between service classes.

The XFG algorithm requires 3 minutes of execution on a Pentium III 1.5 GHz processor to compute the solution. The algorithm constructs 128 transit routes: 19,462 units of bandwidth are assigned to the 40 (2 × 20) direct routes and 14,955 units of capacity to the transit routes. The routes contain 34,417 units of capacity.

The lengths of the LSPs and their capacity assignments are shown in Table 3. Using the same notation as in Table 1, Table 3 shows that 28,678 units of capacity are assigned to the 120 shortest routes (the shortest routes have normalised length 0) and that 19,462 units of capacity are assigned to the 40 direct routes (the direct routes have un-normalised length 1). It is also seen that 85 % of the routes are constructed on the shortest and the shortest-but-one paths connecting the O-D pairs: 90 % of the capacity is configured on these shortest routes.

Some more results are shown in Table 4. The simple TCP model predicts that the average packet loss probability e over all aggregates is 1.9 %. In more detail, the packet loss probability for aggregates of domestic users is 5.4 %, whereas aggregates of business users, with their faster access links and larger data requests, experience a 0.1 % packet loss probability.

It is also seen that the soft blocking factor α is 3.6 % for domestic users and 1.1 % for business users. The effective rates at which users get new data delivered is 46 kbps for domestic users and 2,026 kbps for business users which result in average download times of 5.2 s and 0.2 s respectively.

The algorithm found 168 LSPs, of which 88 are used by domestic users and 80 by business users. An aggregate is said to have an LSP multiplicity of K if it is supported by K LSPs. The table shows that the LSP multiplicity is higher for domestic users than for business users. As shown in (18), a higher multiplicity will give a higher delay. Domestic users are less sensitive than business users to such a delay because of the lower speeds of their access links. The algorithm exploits this relative insensitivity, which results in different multiplicities for the two service classes. LSP bandwidth is the average bandwidth per LSP and as expected, it is seen that business class users consume more bandwidth, partly because of the longer files and partly because of their faster access links. The last observation is confirmed by the LSP loads, which are seen to be 37 % and 24 % for domestic and business users respectively.

The buffers at the ingress nodes of the MPLS network can be used to reduce the packet loss probabilities at the expense of increasing the buffer delay. Such a change will also affect the optimal choices of routes and bandwidths for the LSPs. Table 4 shows the effect of increasing the buffer space on the LSPs for domestic users, which suffer from a high packet loss, from 5 to 10.

The result is not only a lower packet loss probability for domestic users, but the soft blocking factors decrease and the effective rates increase for both types of users. The latter is a consequence of the fact that a complete redesign is

Table 2 Request rate matrix for both domestic users and business users

Node	1	2	3	4	5	6	7	8
1	–	6	7	1	9	5	2	3
2	7	–	24	3	31	15	6	9
3	8	25	–	4	37	18	7	11
4	1	3	3	–	4	7	1	1
5	11	33	39	5	–	24	9	15
6	5	14	16	2	21	–	4	6
7	2	5	6	1	8	4	–	2
8	3	8	10	1	12	6	2	–

free to find completely new LSPs.[2] The lower packet loss probabilities enable faster downloads which, in turn, cause a lower demand for bandwidth from domestic users, hence more bandwidth is made available to business users. This is clearly demonstrated by the fact that the LSP bandwidth goes down and the LSP load goes up for domestic users while the situation is reversed for business users.

Route length	Bandwidths		LSPs	
	C_i	C'_i	L_i	L'_i
0	28,678	0	120	0
1	2,382	19,462	23	40
2	2,808	6,320	17	53
3	549	3,239	8	36
4	0	2,742	0	21
5	0	1,223	0	14
6	0	1,431	0	4
Total	34,417	34,417	168	168

Table 3 Distribution of the allocated bandwidth C_i on LSPs of normalised length i; distribution of the allocated bandwidth C'_i on LSPs of un-normalised length i; distribution of the normalised L_i and un-normalised L'_i LSP lengths

6 Mixed Objective Functions

The results above may be generalised to multi-service networks by applying class dependent objective functions. As a concrete example, consider a UMTS core network. The data carried by such a network can be classified into control traffic and user traffic. Control traffic relates to the signalling required to establish and release connections, to handle mobility etc., and user traffic relates to the four basic service classes in UMTS, viz. conversational, streaming, interactive and background.

Our method to design LSPs can thus be applied to UMTS core networks built on MPLS by mapping the five service classes to class-dependent objective functions as follows:

Signalling and other management information may be carried over a UDP-like protocol which can be represented by an M/G/1/*K* model. The revenue function in this case is e.g. a function of the number of signals transported within certain time limits.

Conversational mainly comprises services like voice for which the Erlang-B based revenue function of Section 4 may be appropriate. Note that this does not imply circuit switching, but only that there is a CAC mechanism in place. It is emphasised that the objective function (2) may be non-linear or linear depending on the specific aggregate considered. Typically, voice will be AMR coded between MSs and TRCs, while standard PCM coding will be used between TRCs and gateways to external networks. The variable bit rate of AMR may be modelled by a non-linear capacity function, whereas the peak rate of PCM implies a linear capacity function.

Streaming is intended for playback of audio and video information. Typically the information will be coded at variable bit rate but the transmission can be carried out at a fixed rate, in particular if the content is known and analysed in advance. Again the Erlang-B based revenue function may be useful, and again this does not imply circuit switching, but only that there is a CAC mechanism in place. Moreover, if bandwidth requirements differ between users and/or contents, extensions to the multi-dimensional version of Erlang-B may be used, e.g. [10, TD9713] and [24].

Interactive may typically carry transaction data of type request-response for which some real time constraints apply. This class may therefore be modelled by a TCP fixed point approach like the one in Section 5.1 with small files and a high tendency for users to be discouraged by slow response times.

Background will be used by non urgent data such as email, etc. Again a TCP fixed point approach may be suitable, although background file sizes are expected to be longer and users will be much less sensitive to response times.

Table 4 Packet loss, session loss, number of paths, path multiplicity and path bandwidth

Service class	All	Dom.	Bus.	All	Dom.	Bus.
Buffer size		5	5		10	5
Packet loss (%)	1.9	5.4	0.1	0.9	2.5	0.1
Request “loss” (%)		3.6	1.1		2.0	0.6
Effective rate (kbps)		46	2026		47	2036
Download time (s)		5.2	0.2		5.1	0.2
LSPs	168	88	80	172	94	78
LSP multiplicity		1.6	1.4		1.7	1.4
LSP bandwidth		108	311		86	376
LSP load (%)		37	24		53	24

[2] *An alternative to complete redesigns is partial redesigns where some LSPs, for example those of business users, are “locked”.*

It is thus seen that the XFG algorithm would in this case use three objective functions: an objective function based on the M/G/1/K queue to compute the price of bandwidth for signalling; the Erlang-B formula to compute the price of bandwidth for conversational and streaming traffic; and an objective function based on the TCP fixed point of Section 5.1 to compute the price of bandwidth for interactive and background traffic. The bandwidth market would use the three objective functions to find and capacitate the LSPs based on the profits to be earned by carrying the various service classes.

In addition it may e.g. be preferred to distinguish network management traffic and, in a split architecture, traffic related to the communication between servers (MSCs, TMSCs, SGSNs and GGSNs) and gateways (MGWs). These two classes would be handled by the same type of objective function as signalling, but with different performance requirements. Typically, the network management information may be relatively insensitive to delays, whereas server-gateway traffic may have strict real time requirements.

From a conceptual point of view, the LSPs can be viewed as members of logically independent networks of which there is one for each service class. The logical networks may for example include RNCs, MSCs and TMSCs for conversational services; RNCs, SGSNs and GGSNs for interactive services; and RNCs, MSCs, TMCSs, SGSNs, GGSNs, HLRs, EIRs etc. for signalling traffic.

The latter observation indicates that the method can be extended to include VPNs in a straightforward manner. The procedure is to characterise the requirements on VPN links by objective functions and represent each VPN link by an aggregate after which XFG will find suitable routes and bandwidths. Also note that the method can be applied on top of existing VPNs.

7 Conclusions

MPLS represents a promising way of obtaining the benefits of different technologies such as ATM (with emphasis on quality of service) and IP (with emphasis on simplicity). In particular, we have shown that multiple LSPs between ELRs which separate not only O-D pairs but also service classes in most realistic cases are likely to be the preferred mode of operation.

We next considered path selection and bandwidth allocation in multi-service MPLS networks in order to optimise the network quality of service. The optimisation was based upon the constrained optimisation of non-linear objective functions. Two examples of such functions were shown in detail, an Erlang-B based one for services carried by e.g. UDP and where equivalent bandwidth applies and a CAC mechanism is active, and another one based on best effort services over TCP where the transmission rate is adapted to the level of congestion. We also discussed the possibilities to include other functions to suit, e.g., services carried by UDP but for which CAC does not apply.

We presented a computationally efficient algorithm called XFG to find and capacitate optimal LSPs. The algorithm is based on a bandwidth market where bandwidth prices determine the routes and bandwidths of LSPs. The algorithm was applied to compute optimal LSPs for a 55 node network model carrying 6 service classes and for an 8 node network carrying 2 service classes. It was seen that the method is capable of quickly designing complex networks of LSPs with appropriate quality of service discrimination between different service classes and that also accounts for node performance characteristics.

A UMTS core network was used as a concrete example to show how the method can be applied to design general multi-service networks, and the extension to include VPNs was described.

The method can be rephrased to distributed design. A first approach is outlined in [8] and further work is in progress. Other points of study include cost based pricing to model design of MPLS networks on leased lines. Finally, work on a more elaborate TCP model which includes short file transfers, slow start, time-outs and a more advanced queuing model is currently in progress.

References

1 IETF. Rosen, E et al. *Multiprotocol Label Switching Architecture.* 2001. (RFC 3031.)

2 IETF. Awduche, D et al. *Requirements for Traffic Engineering over MPLS.* 1999. (RFC 2702.)

3 IETF. Andersson, L et al. *LDP Specification.* 2001. (RFC 3036.)

4 Aboul-Magd, O et al. *Constraint-Based LSP Setup Using LDP.* 2001. IETF draft.

5 Ashwood-Smith, P et al. *Generalized MPLS Signaling – RSVP-TE Extensions.* 2001. IETF draft.

6 IETF. Davie, B et al. *MPLS using LDP and ATM VC Switching.* 2001. (RFC 3035.)

7 Arvidsson, Å. High Level B-ISDN/ATM Traffic Management in Real Time. In: *Perf. Modelling and Evaluation of ATM Networks.* D Kouvatsos (Ed.). London, Chapman and Hall, 1995, 873–878.

8 Berezner, S et al. Local Reconfiguration of ATM Virtual Path Connection Networks. In: *IFIP TC6/WG6.2 Proceedings 5th International Conference on Broadband Communications.* D Tsang, P Kühn (Eds.). Kluwer Academic Publishers, 1999, 621–630.

9 Larsson, S-O. *Capacity Management with Logical Links.* Department of Telecommunications and Signal Processing, Blekinge Institute of Technology, Sweden, 2000. (Ph.D. dissertation.) ISBN 91-628-4293-5.

10 Tran-Gia, P, Vicari, N (Eds.). *Impacts of New Services on the Architecture and Performance of Broadband Networks.* University of Würzburg, Germany, COST-257 Final Report, 2000. ISBN 3-930111-10-1.

11 Weinstein, C, Malpass, M, Fisher, M. Data Traffic Performance of an Integrated Circuit and Packet-Switched Multiplex Structure. *IEEE Trans. on Commun.,* 6 (28), 873–878, 1980.

12 Arvidsson, Å et al. *Cost-Effective Deployment of Bandwidth Partitioning in Broadband Networks.* Submitted.

13 Anerousis, N, Lazar, A. Virtual Path Control for ATM Networks with Call Level Quality of Service Guarantees. *IEEE ACM Trans. on Networking*, 6 (2), 222–236, 1998.

14 Arvidsson, Å, Berezner, S, Krzesinski, A. The Design and Management of ATM Virtual Path Connection Networks. In: *Proceedings of the 7th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'99).* College Park, Maryland, USA, 1999, 2–9.

15 Arvidsson, Å et al. The design of ATM Virtual Path Connection Networks with Service Separation. In: *Proc. 8th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'00).* San Francisco, California, USA, 2000, 424–431.

16 Mitra, D, Morrison, J, Ramakrishnan, K. ATM Network Design and Optimization: a Multirate Loss Network Framework. In: *Proc. of IEEE Infocom '96.* San Francisco, California, USA, 1996, 994–1003.

17 Roughan, M, Erramilli, A, Veitch, D. Network Performance for TCP Networks, Part I: Persistent Sources. In: *Proc. 17th International Teletraffic Congress.* Salvador, Bahia, Brazil, 2001. To appear.

18 Mattis, M et al. The Macroscopic Behaviour of the TCP Congestion Avoidance Algorithm. *Computer Communication Review*, 3 (27), 67–82, 1997.

19 Padhye, J et al. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In: *Proc. ACM Sigcomm '98.* Vancouver, Canada, 1998, 303–314.

20 Cardwell, N, Savage, S, Anderson, T. Modeling TCP Latency. In: *Proc. IEEE Infocom '00.* Tel Aviv, Israel, 2000, 1742–1751.

21 Arvidsson, Å, Krzesinski, A. The Design of MPLS Networks for Optimal TCP Transport. In: *Proc. 4th South African Telecommunications, Networks and Applications Conference.* Margate, South Africa, 2001.

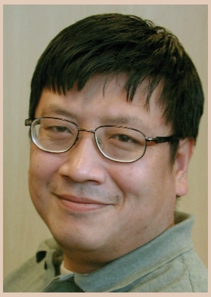
22 IETF. Stevens, W. *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms.* 1997. (RFC 2001.)

23 Stevens, W. *TCP/IP Illustrated, Volume 1.* Reading, Massachusetts, Addison-Wesley, 1999.

24 Berezner, S, Krzesinski, A. An Efficient Stable Recursion to Compute Multiservice Blocking Probabilities. *Performance Evaluation,* 2–3 (43), 151–164, 2001.

State-of-the-art of IP Routing

BONING FENG, ANNE-GRETHE KÅRÅSEN,
PER THOMAS HUTH AND BJØRN SLAGSVOLD



Boning Feng (41) is Research Scientist at Telenor R&D, Kjeller. He is working in the Internet Network Architecture group with special interests in traffic engineering, IP, routing, and simulation. Before he joined Telenor in 1998, he was working as Associate Professor at the Norwegian University of Science and Technology, where he also received his Master and PhD degrees in Telematics with special focus on traffic modelling and analysis.

boning.feng@telenor.com

Anne-Grethe Kåråsen (42) is Research Scientist at Telenor R&D, Kjeller. She is working in the Internet Network Architecture group with special interest in layer 1-3 network management and control.

anne-grethe.karasen @telenor.com

This paper summarises the state-of-the-art of IP routing. Starting with an overview of the currently used routing strategies and protocols in IP networks, it identifies the problems and challenges introduced by the explosive growth of the Internet and the introduction of newer applications and services.

The routing problems in future IP networks are not trivial, with many complex problems yet to be identified and solved. In this paper, a class of routing systems that compute routes subject to satisfaction of a set of constraints and requirements (called “constraint-based routing”, CBR) is presented.

Also included in this paper is a discussion of requirements for developing new routing protocols that support new types of services, such as multicast and mobility.

1 Introduction

As the Internet is growing from a playground for an elite group of computer scientists in the early 80s to a huge network connecting tens (or hundreds) of millions of users, the importance of routing has become more and more obvious. Routing, which is the process of finding a path from a source to every destination in the network, is the underlying structure that glues the world-wide Internet together.

The concept of routing was introduced with the telephone network. Over the past hundred years, many different routing policies have been used in the telephone network. As computerised switching systems have been introduced, the routing policies have become increasingly sophisticated. However, all telephone routing policies have their special features that differ from the IP (or Internet) routing, so we cannot simply adopt these policies to IP networks.

Some of the special characteristics of IP routing compared to routing in telephone networks are listed below:

1. In IP routing, the traffic pattern is less predictable compared to that of the telephone traffic.
2. Routers and links in IP networks are not as reliable as switches and links in the telephone network, so maintaining connectivity is an important task of IP routing.
3. In the Internet, network administrators in different domains may choose different policies, making traffic measurement and management policies much more difficult.
4. Since voice calls require the same, simple quality of service, the admission control decision is trivial. In the Internet, however, connectivity is not sufficient to complete a call: the path must also have sufficient resources available.
5. IP routing is performed on a packet level, while the telephone network routing problem is performed in a circuit switched network.

In this document, we start with an overview of the routing protocols that are currently in use in the Internet (Chapter 2). The increasing traffic demand and new requirements (such as QoS) call for more sophisticated routing protocols. Therefore, in Chapter 3 we discuss a new concept of routing, namely Constraint-based routing. It refers to a class of routing systems that compute routes through a network subject to satisfaction of a set of constraints (e.g. resource availability, policy) and requirements (such as QoS). In the most general setting, constraint-based routing may also seek to optimise overall network performance while minimising costs.

Multi-Protocol Label Switching (MPLS) is a technique promoted by the IETF that integrates the label swapping paradigm with network layer routing. The MPLS ability to support constraint-based routing makes it highly relevant for this study. Therefore, routing in MPLS is discussed in Chapter 4.

In Chapter 5 we give a brief overview of the status and trends of routing research and development activities. Finally some concluding remarks are given in Chapter 6.

2 Current IP Routing Protocols

In this chapter we make an overview of currently deployed routing protocols. We start with a general introduction of routing functionalities, protocol requirements, and choices in the design of routing protocols. Next follows a description of routing in IP networks. We shall discuss the most popular routing protocols that are in use today, and the functionalities that are already implemented there. Also mentioned are new requirements that cannot be satisfied with today's routing protocols.

2.1 Routing Basics

2.1.1 Routing Functionalities

Although different networks employ different routing algorithms, they all share a core of basic routing functionality [STEE95]. The first of the core routing functions is *collecting network and user traffic state information* that is used in generating and selecting routes, and keeping it up to date. The state information includes service requirements and current locations of users, services provided by and resources available within the network, and restrictions on the use of these services and resources.

The second core routing function is *generating and selecting feasible and even optimal routes based on user and network state information*. Feasible routes are those that satisfy all the user- and network-imposed service constraints. Optimal routes are feasible routes that are "best" with respect to a specific performance objective.

Forwarding user traffic along the routes selected was defined as another core routing function. In recent years, however, the term forwarding is classified as a separate function, while routing is now used to describe the first two functions [BLAC00].

2.1.2 Routing Protocol Requirements

Routing protocols are the protocols that establish mutually consistent routing tables in every router in the network. The manner in which the route is calculated is based on a routing algorithm, and the algorithm is a very important part of the overall routing architecture.

Some design goals can be established for routing algorithms [THOM98]:

- **Simplicity**: Since route management is an overhead component in a router, it must not consume too much overhead. As far as possible, routing algorithms should be simple, and they should not consume a lot of memory and CPU capacity.

- **Robustness**: During periods of unusual types of traffic or large volumes of traffic, they should not fail. If they fail, it should not mean a complete loss of routing capacity. The goal of robustness is one aspect of the goal of accuracy.

- **Convergence**: Once a change occurs that requires a route recalculation, the update messages and resulting recalculation of the routes is done quickly, and all nodes reach agreement (convergence) quickly.

- **Flexibility**: A routing algorithm should accommodate different metrics; it should support default routes; it should allow a hierarchy of routing domains, it should support one or more than one path to a destination, etc.

- **Accuracy**: It makes little difference if the route-calculation algorithm is simple, robust, or whatever, if it does not calculate and select accurate routes according to the "best" routing criteria. Of course, the best route depends on the metrics and the algorithm's use of the metrics.

2.1.3 Choices in Routing Protocol Design

Designers of routing protocols have many mechanisms available to them. In this section, we will describe some commonly available choices for routing [MEEL90]. These choices also represent a rough taxonomy to categorise routing protocols.

- *Centralised vs. distributed routing:* In centralised routing, a central processor collects information about the status of each link (up or down, utilisation, capacity, etc.) and processes this information to compute a routing table for every node. It then distributes these tables to all the routers. In distributed routing, routers co-operate using a distributed routing protocol to create mutually consistent routing tables. Centralised routing is reasonable when the network is centrally administrated and the network is not too large. However, it creates a single point of failure, and the concentration of routing traffic to a single point.

- *Intra-domain routing vs. inter-domain routing:* The nodes are grouped into regions on different levels. This implies that the nodes have full knowledge of the topological structure within one region, but only a few are responsible for the routing between the regions. A special case here will be the routing between domains on a general basis, e.g. between autonomous systems (AS).

- *Source-based vs. hop-by-hop:* A packet header can carry the entire route (that is, the address of every router on the path from the source to the destination), or the packet can carry just the destination address, and each router along the path can choose the next hop.These alternatives represent extremes in the degree to which a source can influence the path of a packet. A source route allows a sender to specify a packet's path precisely, but requires

Per Thomas Huth (48) is Research Scientist at Telenor R&D, Kjeller, He is working in the Internet Network Architecture group with special interests in traffic engineering, IP, service differentiation, network utilisation and simulations.

per-thomas.huth@telenor.com

Bjørn Slagsvold (65) is Research Scientist at Telenor R&D and a member of the Internet Network Architecture group, presently working on optical transmission and routing.

bjorn-johan.slagsvold @telenor.com

the source to be aware of the entire network topology. If a link or a router along the path goes down, a source-routed packet will not reach its destination. Moreover, if a path is long, the packet header can be fairly large. Thus, source routing trades off specificity in routing for packet-header size and extra overhead for control messages.

An intermediate solution is to use a *loose source route*. With loose source routes, the sender chooses a subset of routers that the packet should pass through, and the path may include routers not included in the source route. Loose source routes are supported in the IP version 4 and 6 headers.

- *Stochastic vs. deterministic:* With a deterministic route, each router forwards packets toward a destination along exactly one path. In stochastic routing, each router maintains more than one next hop for each possible destination. It randomly picks one of these hops when forwarding a packet.The advantage of stochastic routing is that it spreads the load among many paths, so that the load oscillations characteristic of deterministic routing are eliminated. On the other hand, a destination may receive packets along the same connection out of order, and with varying delays. Consequently, modern networks usually use deterministic routing.

- *Single vs. multiple path:* In single-path routing, a router maintains only one path to each destination. In multiple-path routing, a router maintains a primary path to a destination, along with alternative paths. If the primary path is unavailable for some reason, routers may send packets on the alternative path[1].

- *State-dependent (dynamic) vs. state-independent (static):* With state-dependent or dynamic routing, the choice of a route depends on the current (measured) network state. For example, if some links are heavily loaded, routers may try to route packets around that link. With state-independent or static routing, the route ignores the network state. For example, a shortest-path route (where we measure the path length as the number of hops) is state-independent. State-dependent routing usually finds better routes than state-independent routing, but can suffer from problems caused by network dynamics (such as the routing oscillations). It also requires more overhead for monitoring the network load.

- *Link-state routing vs. distance-vector routing:* In link-state routing each node knows the topology and the cost of each link. In distance-vector routing the vector contains information of topology and cost from the originating node to the destination.

Having broadly considered the choices in routing protocol design, we are going to study specific routing protocols that make a selection from the choices described earlier. The literature on routing is vast. In this paper we only study routing in IP-networks (i.e. the Internet).

2.2 Status of Internet Routing

We refer to the previous discussion (section 2.1.3) of routing protocol choices. In summary, in today's Internet the following choices have been made:

- Centralised routing is not used in the Internet due to the problems of dependability and scalability. The distributed routing approach is implemented.

- Separate protocols are used for intra-domain routing and inter-domain routing, respectively (more discussion in section 2.4).

- Hop-by-hop routing is the most common approach in Internet today. However, it is believed that source routing (or explicit routing) might be better suited to select paths satisfying user's QoS requests [SU00]. Therefore, research on source routing has got more focus in recent years.

- Deterministic routing is preferred due to its ability to offer more consistent quality of service.

- Currently, single-path routing is used on the Internet, because maintaining alternative paths requires more routing table space. However, multiple-path routing can reduce both the packet blocking probability and the restoration time in the presence of failure. Multiple-path routing can also give better support to the QoS requirements that are becoming more and more important for the future Internet.

- The Internet uses both state-dependent and state-independent routing.

- Both distance-vector and link-state routing are used, as discussed in the next section.

[1] *With stochastic routing, routers may send packets on alternative paths even if the primary path is available.*

2.3 Distance-vector and Link-state Routing

The two fundamental routing algorithms in IP networks are distance-vector and link-state routing.

Both algorithms assume that a router knows

- the address of each neighbour, and

- the cost of reaching each neighbour (where the cost measures quantities like the link's capacity, the current queuing delay, or a per-packet charge).

Both algorithms allow a router to find *global* routing information, that is, the next hop to reach every destination in the network by the shortest path, by exchanging routing information with only its neighbour. In the following, we will give a brief introduction to these two algorithms. More details can be found in many places in the literature, such as [HUIT00].

2.3.1 Distance-vector Routing

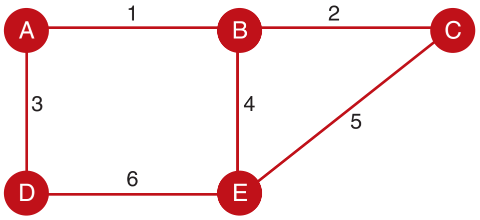
In distance-vector routing, we assume that each router knows the identity of every other router in the network (but not necessarily the shortest path to it). Each router maintains a *distance vector*, which is a list of <*destination, cost*> tuples, one tuple per destination, where cost is the current estimate for the sum of the link costs on the shortest path to that destination. Each router initialises the cost to reach all non-neighbour nodes to a value higher than the expected cost of any route in the network (commonly referred to in the routing literature as *infinity*). A router sends a copy of its distance vector to all its neighbours. When a router receives a distance vector from a neighbour, it determines whether its cost to reach any destination would decrease if it routed packets to that destination through that neighbour (Figure 2.1). It can easily do so by comparing its current cost to reach a destination with the sum of the cost to reach its neighbour and its neighbour's cost to reach that destination.

With the continued exchange of distance vectors, the cost of every link is eventually known throughout the network. The distance-vector algorithm is also called *Bellman-Ford* ([BELL57], [FF62]) after its creators.

The distance-vector algorithm works well if nodes and links are always up, but it runs into many problems when links go down or come up. The root cause of problems is that when a node updates and distributes a distance vector, it hides the sequence of operations it used to compute the vector. Thus, downstream routers do not have sufficient information to figure out whether their choice of a next hop will cause loops to form. One typical problem is *count-to-infinity* [KESH97]. Solutions to this problem can be found in many places in the routing literature (e.g. [HUIT00]).

2.3.2 Link-state Routing

In distance-vector routing, a router knows only the cost to each destination or, sometimes, the path to the destination. This cost of path is partly determined on its behalf by other routers in the network. This hiding of information is the cause of many problems with distance-vector algorithms.



Initial

	A	B	C	D	E
A	0	1	∞	3	∞
B	1	0	2	∞	4
C	∞	2	0	∞	5
D	3	∞	∞	0	6
E	∞	4	5	6	0

Computation at A when Distance Vectors from B and D arrive

1. *Cost to destinations via B = Cost to go to B + Cost to destinations from B = (1,1,1,1,1) + (1,0,2,∞,4) = (2,1,3,∞,5)*

2. *Cost to destinations via D = Cost to go to D + Cost to destinations from D = (3,3,3,3,3) + (3,∞,∞,0,6) = (6,∞,∞,3,9)*

3. *Current cost from A = (0,1,∞,3,∞)*

Minimum cost to destinations = (0,1,3,3,5)

Figure 2.1 Distance-vector algorithm at node A. In the figure, A receives a distance vector from its neighbours B and D. It uses this information to find that it can reach nodes C and E at a lower cost. It therefore updates its own distance vector and chooses B as its next hop to nodes C and E

In contrast, the philosophy in link-state routing is to distribute the topology of the network and the cost of each link to all the routers. Each router independently computes optimal paths to every destination. If each router sees the same cost for each link and uses the same algorithm to compute the best path, the routes are guaranteed to be loop free. Thus, the key elements in link-state routing are a way to distribute knowledge of network topology to every router in the network, and a way to compute shortest paths given the topology.

Topology Dissemination

Each router participating in the link-state algorithm creates a set of *link-state advertisements* (LSAs) that describe its links. An LSA contains the router's ID, the neighbour's ID, and the cost of the link to the neighbour. The next step is to distribute a copy of every LSA to every router using *controlled flooding*. The idea is that when a router receives a new LSA, it stores a copy of the LSA in a *link state database*, and forwards the LSA to every interface other than the one on which it arrived.

Computing Shortest Paths

A router typically uses *Dijkstra's shortest-path first algorithm* [DIJK59] to compute optimal routes in the network. A good description of the algorithm is given in [KESH97].

When the algorithm stops, we have, for each router, the route on the shortest path used to reach it.

Complexity

Generally, link-state algorithms are complex. Much overhead is needed in order to prevent corruption of the Link State Databases and keep them coherent. It also requires that nodes independently compute consistent routes. In large networks, LSAs also require much memory in the routers.

2.3.3 Link-state versus Vector-distance

Both link-state and vector-distance routing are commonly used in the Internet today. Distance-vector routing does not require that nodes independently compute consistent routes. They also require less memory for routing tables than do link-state protocols, because they do not need to maintain a link-state database.

Vector-distance routing was introduced first with *Routing Information Protocol* (RIP), which is a very simple protocol. It works well with small networks. However, for large and complex networks RIP is probably wholly inadequate:

- It does compute new routes after any change in network topology, but in some cases it does so slowly, by counting to infinity.

- RIP cannot be used in networks in which routes may use more than *15* hops, because a metric of *16* indicates infinity.

  On the other hand, conventional wisdom is that link-state routing is more stable because each router knows the entire network topology. The advantages of link-state routing are, among others:

- Fast, loopless convergence;

- Support for precise metrics and, in the future, multiple metrics;

- Support of multiple paths to a destination.

The focus of research on routing in recent years has been on link-state routing. Although the protocols are more complex, the extra functionalities they offer can be very useful to support new service requirements in modern IP-networks.

2.4 Hierarchical Structures & Domains

2.4.1 Autonomous Systems

Since the beginning of the Internet, the concept of *Autonomous System* (AS) has been used to define a set of routers and networks under the same administration.

In the early days of the Internet, the network consisted of a small number of campus networks that were interconnected via one single backbone network (which is also called the "*core*").

From a routing point of view, the definition of an AS is quite simple: all parts of an AS must remain connected [HUIT00]. Routers belonging to the same AS must exchange routing information in order to maintain connectivity. This is normally achieved by selecting a single routing protocol and running it between all the routers. Therefore, a consistent internal routing policy is employed within an AS.

2.4.2 Interior Gateway Protocols

Routing protocols employed within ASs are called *Interior Gateway Protocols* (IGPs). The most used IGPs include RIP, OSPF, and IS-IS.

Splitting the Internet into several ASs aims at lowering the routing overhead and at easing the network management. Computing routes, distributing new versions of software, or isolating failing elements is easier when the number of links and routers is kept relatively small. However, connectivity must be maintained. The routing tables inside the AS should include entries covering all possible Internet destinations. Since IGP is only used within an AS, the choice of IGP in one AS is independent of that of another AS.

The routing tables are maintained by the IGP, but the IGP messages are only exchanged between routers that belong to the AS. These routers can only discover information about the internal networks to which they are directly connected. They must get the information about the exterior networks through a dialogue with exterior gateways, which are entry points in adjacent autonomous systems.

2.4.3 Exterior Gateway Protocols

The role of *Exterior Gateway Protocol* (or simply called *Exterior Protocol*) is precisely to exchange this "reachability information" in order to enable the ASs to exchange routing information.

Although all the routers within an AS are mutually co-operative, routers interconnecting two ASs may not necessarily trust each other. Exterior protocols determine routing between entities that can be owned by mutually suspicious domains. An important part of exterior protocols, therefore, is configuring *border gateways* (that is, gateways that mediate between interior and exterior routing) to recognise a set of valid neighbours and valid paths.

The EGP-protocol was designed for this purpose. Although EGP is still in use today, however, it is being replaced by *Border Gateway Protocol* (BGP). The BGP version that has been in use since 1995 is BGP-4 [RFC1771].

BGP-4 is a path-vector protocol, where distance vectors are annotated not only with the entire path used to compute each distance, but also with certain policy attributes. It guarantees loop-freeness at the expense of large routing tables. BGP routers use TCP to communicate with each other, instead of layering the routing messages directly over IP, as is done in every other Internet routing protocol. This simplifies the error management in the routing protocol. However, routing updates are subject to TCP flow control, which can lead to fairly complicated and poorly understood network dynamics. For example, routing updates might be delayed waiting for TCP to time out. Thus, the choice of TCP is still controversial [KESH97].

If an AS has more than one BGP-speaking border gateway, path vectors arriving at a gateway must somehow make their way to all the other gateways in the AS (also called internal peering). BGP-4 is hard to maintain because of the need to choose consistent path attributes from all the border routers, and to maintain clique connectivity among internal peers.

2.4.4 Interconnecting Exterior and Interior Routing Protocols

The key problem in interconnecting exterior and interior protocols is that they may use different routing techniques and different ways to decide link costs. For example, the exterior protocol may advertise a 5-hop count to another AS. However, each of these hops may span a continent and cannot be compared with a 5-hop path in the interior of an AS.

A similar problem arises if the interior and exterior routing protocols use different routing schemes. For example, the exterior protocol may use path-vector routing, and the interior may use link-state routing. Thus, the border gateway must convert from a link state database to a set of distance vectors that summarise paths to its interior. In the other direction, it must convert from distance-vector advertisements to external records for the interior routing protocol.

The bottom line is that interconnecting a given interior and exterior protocol requires a fair amount of manual intervention, and frequent monitoring to ensure that the network stays up. This is a direct consequence of the heterogeneity in the administration of the Internet, and of its decentralised control.

2.5 Supporting QoS Requirements

Routing deployed in today's Internet typically supports only one type of datagram service called "best-effort". No guarantee of QoS requirements is offered, but routing is optimised for a single arbitrary metric, administrative weight or hop count. Alternative paths with acceptable but non-optimal cost cannot be used to route traffic.

In order to support integrated-services class of services, multiple paths between node pairs will have to be calculated. Some of these new classes of service will require the distribution of additional routing metrics, e.g. delay, and available bandwidth. If any of these metrics change frequently, routing updates can become more frequent, thereby consuming network bandwidth and router CPU cycles.

A second problem is that today's routing will shift the traffic from one path to another as soon as a "better" path is found. The traffic will be shifted even if the existing path can meet the service requirements of the exiting traffic. If routing calculation is tied to frequently changing consumable resources (e.g. available bandwidth) this change will happen more often and can introduce routing oscillations as traffic shifts back and forth between alternate paths. Furthermore, frequently changing routes can increase the delay variation (jitter) experienced by the end users.

A third problem with today's routing is that if the existing path cannot admit a new flow, the associated traffic cannot be forwarded even if an adequate path exists.

2.6 Need for better Routing Protocols

The Internet is growing fast – the number of connected hosts is doubled almost every year, while the volume of traffic is doubling every six to ten months. This growth has been sustained for several years, and all measures indicate that it may well continue at the same rate for several years, [HUIT00].

Internet providers must invest continuously to build up network capacity, but they also have to cope with another problem – as the Internet grows, the number of routers that have to be propagated by routing protocols also grows, resulting in more routing traffic.

Let us look at one example: a problem with BGP is that a small fraction of the routes contributes an inordinate amount of updates. This phenomenon, informally known as "route flap", can be caused by software or hardware bugs, by the interaction between BGP and network congestion described in the previous section, or by local decisions. Whatever the cause, it is necessary to mitigate its effects. If a misbehaving router sends too many updates at too short intervals, its neighbours that try to process all the updates will exhaust their computing resource, and may fall into a congested state that triggers further instabilities.

One solution, proposed in [RFC2439], is to limit the rate at which updates are accepted for any given path.

The problems mentioned above are only examples of what can happen in a large Internet. There are many other challenges in the area of routing. Some of these are:

- Problems related to interconnection between ASs;

- How to manage large ASs – since IGP requires that all the routers know each other within the AS, a good idea is to divide the AS into several small "sub domains";

- How the routing tables can be aggregated to cope with the growth of the Internet routing table;

- Is IPv6 the solution?

In the Internet today, routing is typically done in a distributed fashion. Routes are optimized for a single arbitrary metric, administrative weight, or hop count. For any source-destination pair, all the packets follow the current "shortest path" (i.e. lowest cost path). Alternatively, fully acceptable routes are not used if they represent higher "cost"[2)].

The current IP routing protocols were designed for "elastic traffic", such as TCP based applications like FTP, HTTP, etc., which are insensitive to delay and delay-variations.

In order to support the traffic growth and new types of services that are planned to be transported over IP-networks and the corresponding QoS-requirements, we need better routing protocols.

3 Constraint-based Routing

Constraint-based routing refers to a class of routing systems that compute routes through a network subject to satisfaction of a set of constraints and requirements. In the most general setting, constraint-based routing may also seek to optimise overall network performance while minimising costs.

The constraints and requirements may be imposed by the network itself or by administrative policies. Constraints may include bandwidth, hop count, delay, and policy instruments such as resource class attributes. Constraints may also include domain specific attributes of certain network technologies and contexts that impose restrictions on the solution space of the routing function. Path oriented technologies such as MPLS have made constraint-based routing feasible and attractive in public IP networks.

Constraint-based routing is in general applicable to traffic aggregates as well as flows, and may be subject to a wide variety of constraints that may include policy restrictions.

[2)] *Some routing protocols, such as OSPF, do support alternative routes with equal cost, so a split of traffic among several equal-cost paths are accepted.*

3.1 Definition of Constraints

Constraints and resources are counterparts to one another: routes have constraints while network elements (nodes and links) have resources. As paths are explored, the constraints for a route are checked against the resources along the path to see that the constraints are met. The constraints specified must match available resource information [CR-notes].

Constraints can be divided into Boolean and quantitative. Some constraints can be of both types. Boolean constraints indicate whether or not a candidate path is feasible. Quantitative constraints assign numerical values to paths, enabling choice between feasible candidate paths.

Resources can be divided into configurable, dynamic and topological. Configurable resources are those assigned by an administrator, e.g. administrative groups and link metrics. Dynamic resources are those that depend on the network state and vary with time, e.g. available link bandwidth. Topological resources are those that are enforced by the topology of the network, e.g. path length.

3.1.1 Boolean Constraints

Boolean constraints include (related resource in brackets):

- Administrative group constraints (administrative groups or colours configured on links);

- Bandwidth availability (available link bandwidth);

- Delay bounds (configured delay on links and nodes);

- Hop count bounds (path length).

3.1.2 Quantitative Constraints

Quantitative constraints include (related resources in brackets):

- Residual bandwidth ratio (residual link bandwidth);

- Path metric (metric);

- Resilience (penalty, applies to backup path computation);

- Hop count (path length).

In order to select a path among feasible candidate paths, the quantitative constraints have to be ordered or prioritised in some way. This order should be administratively configurable. A default ordering of the quantitative constraints could e.g. be: path metric, resilience, residual bandwidth ratio and hop count (suggested by [CR-notes]).

3.2 QoS Routing

A definition of QoS(-based) routing [RFC2386]: A routing mechanism under which paths for flows are determined based on some knowledge of resource availability in the network as well as the QoS requirement of flows.

Another definition of QoS routing [QoSGlos]: A dynamic routing protocol that has expanded its path-selection criteria to include QoS parameters such as available bandwidth, link and end-to-end path utilisation, node resource consumption, delay and latency, and induced jitter.

QoS routing is regarded as a subset of the more general constraint-based routing concept. It selects routes with sufficient resources for requested QoS parameters.

The main objectives of QoS routing are:

- Dynamic determination of feasible paths;
- Optimisation of resource usage.

The QoS requirement of a flow is given as a set of constraints, which can be *link constraints, path constraints* or *tree constraints* (applicable to multicast flows only). A link constraint specifies a restriction on the use of links. A bandwidth constraint of a unicast path will e.g. require that the links constituting the path must have a minimum amount of available bandwidth. A path constraint specifies the end-to-end QoS requirement on a given (single) path, while a tree constraint specifies the QoS requirement for the whole multicast tree. A feasible path is a path that has sufficient unused resources to satisfy the QoS constraints of a flow. The basic function of QoS routing is to find such a path. Additionally, the applied QoS routing algorithm may try to optimise resource utilisation by considering link *cost*. The optimal output of a QoS routing procedure is the lowest-cost path among all feasible paths.

3.2.1 Network State Information

In order to find a feasible path for a new flow, it is necessary to have up-to-date state information. The state information may be classified as described in the following.

Each node is assumed to maintain its up-to-date *local state*, including the queuing and propagation delay, the residual bandwidth of the outgoing links, and the availability of other resources.

The combination of the local state of all nodes in the network is called a *global state*. An IGP with appropriate TE extensions may be used to spread this information among the network nodes so that each node knows the topology of the network and the state of each link as illustrated in Figure 3.1. The information kept by the nodes

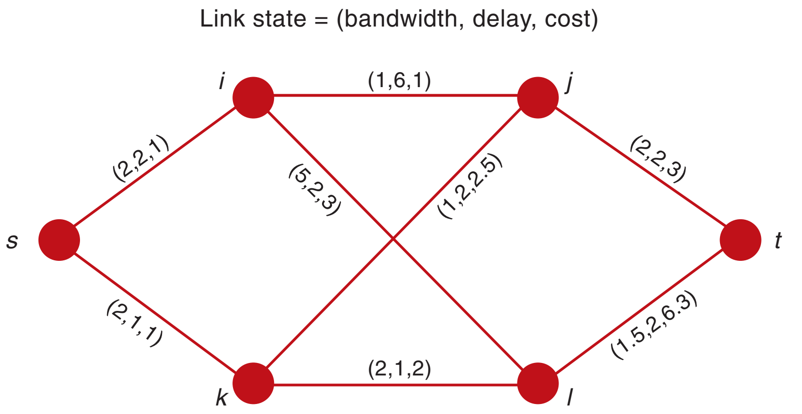


Figure 3.1 Network state [CHEN 98]

will never be completely up-to-date due to the non-negligible delay of the information dissemination process. The larger the network, the more imprecise the information gets.

To reduce the scalability problem for larger networks, information may be aggregated according to the hierarchical structure of the network, obtaining an *aggregated (partial) global state* view. [CHEN98] has described these matters in more detail.

3.2.2 The Unicast Routing Problem

The *unicast routing problem* is defined as follows: given a source node *s*, a destination node *t*, a set of QoS constraints *C*, and possibly an optimisation goal, find the best feasible path from *s* to *t* that satisfies *C*.

QoS requirements on metrics such as residual bandwidth and buffer space, so-called "bottleneck" requirements, are relatively easy to handle. The state of the resulting path is determined by the state of the bottleneck link. In Figure 3.1, the bandwidth of path $s - i - j - t$ is 1, which is the bandwidth of the bottleneck link (i, j). In this case, two basic routing problems may be defined. One problem is called *link-optimisation routing*. For example, bandwidth-optimisation routing is to find a path that has the largest bandwidth on the bottleneck link (the widest path). The other problem is called *link-constrained routing*. For example, bandwidth-constrained routing is to find a path whose bottleneck link has a bandwidth above a required value.

QoS requirements on metrics such as delay and cost, so called "additive" requirements, are more complex to handle. The state of the path is determined by the combined state of all links on the path. In Figure 3.1, the delay of path $s - i - j - t$ is 10, which is the total delay of all links on the path. In this case, two basic routing problems may be defined. One problem is called *path-optimisation routing*. For example, least-cost routing is to find a path whose total cost is minimised. The other problem is called *path-constrained routing*. For example, delay-constrained routing is to find a path whose delay is bounded by a required value.

A number of composite routing problems can be derived from the four basic problems cited above. Some of these composite routing problems are hard to solve. For details, see [CHEN98].

Proposed traffic engineering extensions to OSPF [OSPF-TE] and IS-IS [ISIS-TE] currently support the advertisement of a single routing metric, in addition to bandwidth and resource class information. Additional link metrics, e.g. delay-related metrics, are not supported. Thus, information will not be directly available "on-line" for calculating paths with delay-related QoS requirements. A solution to get around this might be to use the resource class information (link colour) to mark links, so that e.g. satellite links are avoided for delay-sensitive traffic.

There is work in progress that considers IGP extensions supporting multiple metrics, see e.g. [Fedyk].

3.2.3 Path Precomputation versus Dynamic QoS Routing

QoS routing may primarily be aimed at traffic engineering, and the operation is characterised by a long timescale and a coarse granularity of the traffic flow it handles (traffic aggregates). In this case, the goal of QoS routing is to maximise the network performance in the presence of slowly changing traffic patterns. The different paths computed by QoS routing are either pre-established or change only infrequently. Several proposed QoS routing protocols are based on precomputing paths for all possible QoS requirements, and then assign traffic to the paths accordingly. An example would be the establishment of MPLS LSPs to accommodate traffic with varying QoS requirements (e.g. DiffServ classes). A drawback here is that the use of traffic aggregates and the focus on network wide traffic optimisation cannot provide explicit QoS guarantees to individual flows. The precomputation perspective of QoS routing is described in detail in [ORDA00].

The other extreme is to compute QoS routes for each request, where each request explicitly express its resource requirements (e.g. similar to IntServ). The QoS routing will in this case be constrained by satisfying individual QoS requirements, rather than obtaining a more global optimisation of network performance and resource usage.

3.2.4 Routing QoS and Best-effort Traffic

QoS routed and best-effort traffic will coexist in most networks, and this may cause a conflict of interest between the two. If QoS traffic were supported, a goal would be to admit as many QoS flows into the network as possible. At the same time, another goal would be to optimize the throughput and responsiveness of best-effort traffic. Generally speaking, QoS traffic is not affected by best-effort traffic due to resource reservation. However, if the overall traffic in the network is misjudged, the throughput of best-effort traffic will suffer. Links with light QoS traffic may for example have heavy best-effort traffic. These links will often be considered good candidates for additional QoS flows, causing the congested best-effort traffic to become even more congested.

3.3 Policy-based Routing

Policy-based routing is regarded as another subset of the more general constraint-based routing concept.

The most common reason to do policy routing is to accommodate "acceptable-use policies" and to select providers.

The requirement for policy routing appeared with the "commercialisation" of the Internet. Users of the early Internet did not care much about the route that was used for carrying their packets. The network was perceived "free", a "public good" that should simply be shared evenly. But commercial users should not benefit from public subsidies, and thus could not use the "default" route through the "academic" backbones. They had to alter the shortest path to take a policy requirement into account.

The requirement for policies then became more and more sophisticated. Merely finding one acceptable route is not enough when the users are charged for their traffic. A user may e.g. want to switch to another provider between 1:00 PM and 3:00 PM to benefit from better rates.

The principles of policy-based routing are quite similar to those of QoS routing, with differences in the service requirements.

The past attempts at policy routing have not been successful. There are lots of business and technical difficulties that are still not solved. MPLS, which we discuss in Chapter 4, could be a good candidate to implement policy routing successfully.

4 Routing in MPLS

MPLS is a technique promoted by the IETF that integrates the label-swapping paradigm with network layer routing. A label switched path (LSP) is the route that data follows between the ingress and egress of an MPLS domain.

Assigning IP traffic to MPLS hop-by-hop LSPs may improve IP performance since label switching requires less processing than traditional IP forwarding. However, it is the MPLS ability to provide for constraint-based routed LSPs that is expected to be most important to IP traffic engineering. The general concept of constraint-based routing is described in Chapter 3.

In [RFC2702] the "traffic trunk" concept is used. A traffic trunk is an aggregation of traffic flows of the same class, which are placed inside an LSP. A traffic trunk is an abstract representation of traffic to which specific characteristics can be associated. Traffic trunks are routable objects (similar to e.g. ATM VCs). There is a distinction between a traffic trunk and the path (LSP) through which the traffic traverses. A traffic trunk can be moved from one path to another. In practice, the terms LSP and traffic trunk are often used synonymously. The term LSP tunnel is commonly used to refer to the combination of traffic trunk and explicit LSPs in MPLS. In this chapter, the terms LSP and ER-LSP are used, although the term LSP tunnel might be more appropriate in some places.

An MPLS traffic engineering model consists of four basic functional components:

- Network state information dissemination;
- Path management;
- Traffic assignment;
- Network management.

Network state information dissemination and the path selection component of path management are the parts that constitute the routing aspect of MPLS TE.

4.1 Network State Information Dissemination

In support of constraint-based routing, IETF is defining IGP traffic engineering extensions that include link attributes as part of each router's LSA, see e.g. [OSPF-TE] and [ISIS-TE]. Relevant link attributes include:

- Link type;
- Traffic engineering metric;
- Maximum bandwidth;
- Maximum reservable bandwidth;
- Unreserved bandwidth;
- Resource class/colour.

The standard link-state IGP flooding algorithm distributes these additional link attributes to all routers in the routing domain. The edge routers, usually ingress LSRs, use this information, together with traditional topology information and administrative input, for online calculation of LSP paths (see 4.2).

4.2 Path Selection

Paths can be computed automatically by the underlying routing protocols, or they can be defined administratively by a network operator. If there are no resource requirements or restrictions associated with an LSP, then a topology driven protocol can be used to select its path. LSPs routed in this manner are called control-driven or hop-by-hop LSPs. However, if resource requirements or policy restrictions exist, then a constraint-based routing scheme should be used for path selection.

There are a number of ways to route an LSP:

1 The full path for the LSP may be calculated offline;

2 A partial path for the LSP may be calculated offline, permitting online calculation in the ingress router to determine the full path;

3 The full path for the LSP may be calculated online, based on the input of LSP constraints;

4 The full path for the LSP may be calculated online, with no input of LSP constraints.

Cases 1 and 2 are described in 4.2.3, while case 3 is described in 4.2.2. Case 4 above results in normal IGP shortest-path routing for the LSP, and no further description is given here.

4.2.1 The CSPF Algorithm

Constrained shortest path first (CSPF) is a shortest path first (SPF) algorithm that has been modified to take into account specific restriction when calculating the shortest path across the network. Constraint-based routing becomes relatively simple when this algorithm is used. The algorithm seems to be convenient for online path selection, where one LSP path is calculated at a time. However, when multiple LSPs are to be routed, CSPF may have difficulty finding feasible routes even if they exist.

The CSPF algorithm requires input of the type:

- Topology link-state information;

- Attributes associated with the state of network resources (link attributes, see 4.1);

- Administrative attributes required to support traffic traversing the LSP (e.g. bandwidth requirements, maximum hop count, administrative policy requirements).

All candidate nodes and links for a new LSP are considered. CSPF rejects all path components that do not meet the route requirements (constraints). The output of the CSPF calculation is an explicit route consisting of a sequence of LSR addresses that provides *the shortest path that meets the constraints*.

[JUNOS] presents the CSPF implementation from Juniper Networks. Since a description of such detail is lacking from other material that has been studied, their solution may be studied to get a more detailed impression of the CSPF concept.

4.2.2 Online Path Selection

Each router maintains network link attributes and topology information in a database. The information is placed in the database after being flooded by the IGP.

Each ingress router uses this database to calculate the paths for its own set of LSPs across the MPLS domain. The path for each LSP can be represented by either a strict or loose explicit route. If the ingress router specifies all the LSRs in the LSP, the LSP is identified by a strict explicit route. If the ingress router specifies only some of the LSRs in the LSP, the LSP is described by a *loose* explicit route.

The ingress router may apply a CSPF algorithm (see 4.2.1) to the information in the database to determine the LSP paths.

4.2.3 Offline Path Selection

An administratively specified explicit path for an LSP is configured through operator action. The path may be completely specified or partially specified. The path is completely specified if all the hops between the LSP endpoints are identified. The path is partly specified if only some of the hops are identified, leaving the completion of the path selection to online route calculation.

When a path has been fully calculated offline, the LSP may be instantiated in two ways. Each router in the LSP may be individually configured with the necessary static forwarding state. Alternatively, the ingress router may be configured with the full path. The ingress router then uses [CR-LDP] or [RSVP-TE] as a dynamic signalling protocol to install forwarding state in each router along the LSP. The resulting LSP is termed a strict ER-LSP.

When a path has been partly calculated offline, the ingress router may explicitly complete the route calculation and instantiate the LSP by use of signalling. In this case the resulting LSP is termed a strict ER-LSP.

For the parts of the route that have not been calculated offline, the ingress router may also use abstract nodes in the explicit route representation. This permits local flexibility in fulfilling the request for a constraint-based route. The resulting LSP is termed a loose ER-LSP. In this case, the question of route pinning should be considered. Route pinning is applicable to segments of the ER-LSP that are loosely routed, and should be applied if it is undesirable to change the path for the loosely routed segments of the LSP.

If global optimisation of network resources is required, the LSP path selection must take place offline. Online path selection calculates one LSP at a time, and the order in which the LSPs are calculated determines the resulting set of physical paths in the network. This will probably not result in optimal network resource utilisation. An offline path selection tool is able to simultaneously examine the requirements for each LSP and the resource constraints of each link. A global calculation may be performed, and the output of this calculation is a set of LSPs that optimises resource utilisation for the network as a whole. After completion of the offline calculation, the LSPs may be instantiated in any order.

4.2.4 Generic Traffic Trunk Attributes

A traffic trunk is defined as an aggregation of traffic flows of the same class that are placed inside an LSP. This abstract representation of traffic allows for specific characteristics to be associated with traffic aggregates. Traffic trunks are objects that can be routed, and traffic trunk characteristics may put constraints on the path of the LSP into which it is placed.

A number of generic traffic trunk attributes have been defined in [RFC2702]. Some of these attributes are applicable to LSP path selection, and are described below.

Traffic parameters indicate the resource requirements for the traffic trunk, as they define the characteristics of the FEC to be transported through the LSP. These characteristics may include peak rates, average rates, permissible burst size, etc. For the purpose of path selection, or bandwidth allocation in general, the traffic parameters can be used to calculate a single value for the LSP bandwidth requirements.

Resource class affinity attributes associated with a traffic trunk can be used to specify the class of resources that are to be explicitly included or excluded from the path of the traffic trunk, i.e. from the LSP.

The *priority* attribute defines the relative importance of LSPs. Priorities can be used to determine the order in which path selection is done for LSPs. Priorities are also important if pre-emption (see below) is permitted. They can be used to define a partial order on a set of LSPs, and pre-emptive policies may be actualised according to this. [CR-LDP] defines two priority parameters, namely setupPriority and holdingPriority.

The *pre-emption* attribute determines whether a specific LSP can pre-empt another LSP from a given path, and whether another LSP can pre-empt a specific LSP. Pre-emption means rerouting existing LSPs to reallocate resources to a new path. Pre-emption can be used to assure that relatively favourable paths always can be selected for high priority LSPs. Setup and holding priorities are used to rank existing LSPs and the new LSP to determine if the new LSP can pre-empt an existing one.

A *path preference rule* attribute should be associated with administratively specified ER-LSPs. This is a binary attribute with values “mandatory” and “non-mandatory”. If the ER-LSP path is defined as “mandatory”, then that path must be used. If the specified path for some reason cannot be instantiated, the LSP instantiation process fails. If the LSP instantiation process succeeds, the LSP is implicitly pinned. On the other hand, “non-mandatory” paths are used if feasible. If not, an alternate path can be chosen instead by the ingress router.

4.2.5 Generic Resource Attributes

A number of generic resource attributes have been defined in [RFC2702]. Some of these attributes are applicable to path selection, and are described below.

The *maximum allocation multiplier* of a resource is an administratively configurable attribute that determines the proportion of the resource available for allocation to LSPs. The attribute is most applicable to link bandwidth, but can also be applied to buffer resources on LSRs. The relationship between maximum bandwidth and maximum reservable bandwidth of a link represents the maximum allocation multiplier concept.

Resource class attributes can be viewed as "colours" assigned to resources such that the set of resources with the same colour belongs to the same "class". Resource class attributes are administratively assigned parameters, and they can be used to implement various policies. Links are the resources of special interest, and link colour is one of the link attributes included in the IGP TE extensions (see 4.1).

5 Development Trends

5.1 IP Multicast Routing

Today, most Internet applications are so-called *unicast* because they use a point-to-point transmission infrastructure. The usage of point-to-multipoint transmission was limited to local network applications due to the natural broadcast capabilities of LANs. But during the past few years, we have observed the emergence of new applications that use *multicast* transmission to enable efficient communication among a group of hosts (instead of two hosts). These applications require "multicast routing" – sending an IP packet to a "group" address so that it reaches all the members of the group, which may be scattered throughout the Internet.

There is a number of key challenges that must be met by a multicast routing algorithm to be applicable to the Internet. It must route data only to group members, optimise routes from the source to receivers, maintain loop-free routes, and not concentrate all multicast traffic on a subset of links. Furthermore, the signalling in creating and maintaining a group must scale well with a dynamic receiver set.

In order to provide a multicast service, one also has to implement a complex protocol architecture not limited to a single routing protocol. Issues like address allocation, domain isolation, access control, and security have to be provided for multicast to become a commercial service.

Today, multicast has not yet matured enough to be widely used. Research efforts are currently trying to address the scalability issues by providing a simpler architecture. The future of multicast routing relies on these efforts, and also on the capacity of the network providers to define business models that can fund the deployment of the service.

5.2 Mobility

With the advent of portable computers, the need to support mobility in the Internet has become pressing in recent years. According to the IETF mobile IP working group, the requirements for a mobile IP solution are:

1 A mobile host should be capable of continuing to communicate, using the same IP address, after it has been disconnected from the Internet and reconnected at a different point.

2 A mobile host should be capable of interoperating with existing hosts, routers, and services.

The first requirement is dictated by the need to maintain TCP/IP connections while the mobile host is roaming from cell to cell. Keeping a single IP address is essential because this address identifies the TCP connection. The second requirement is dictated by the need for "gradual deployment".

A few other "soft requirements" were also listed by the IETF mobile-IP group.
1 No weakening of IP security;
2 Multicast capability;
3. Location privacy.

The basic model for supporting mobility in the Internet is presented in IETF [RFC2002]. In the same document, the mobile-IP group also defined one single routing protocol. In order to reach an agreement they took many shortcuts.

We are now only seeing the beginning of mobile computing. IP extensions for mobility are being standardized, and the real deployment is slowly starting. The current protocols have been designed in a very conservative fashion, so as to work in the current Internet. Many refinements will have to be addressed in further versions of IP mobility. To gain the advantage of mobility, one will probably have to update the routing protocols so that they will allow multiple home agents and clusters of bases [HUIT00].

6 Conclusion

In this paper we have given an overview of the state-of-the-art of IP routing.

In today's Internet, there is a clear distinction between intra-domain protocols (IGPs) and inter-domain protocols (BGPs).

Each ISP can make its own choice of IGP. The most popular IGPs are still those defined in the 80s (and based on algorithms from the late 50s), although we observe a shift from vector-distance protocols (such as RIP) to link-state protocols (such as OSPF and IS-IS). Link-state protocols have many advantages compared to the vector-distance ones; however, they still have some major drawbacks. One of these is that all the IGPs used in the Internet today offer best-effort service, which means that they do not support QoS requirements.

Inter-domain routing is a much more complicated process. Since it deals with routing between different domains, standardisation is necessary. The most popular protocol today is BGP-4, which was standardised in 1995. BGP-4 incorporates many nice features compared to its predecessor, the EGP. However, the protocol is quite complicated, and the whole process relies heavily on manual configuration of routing parameters in "BGP configuration table". This requires very skilled operators.

With the explosion of Internet traffic, the influence of the market on the technology development has become more pronounced from the mid 90s and onwards. How to improve network performance has been a hot topic among the vendors. Along with the development of hardware technology, research on new routing strategies gained more attention in recent years. The phenomenal growth of the Internet has also led to an increased demand on the network to offer differentiated services. Constraint-based routing seems to be able to support this by setting different constraints and requirements for different classes of services. Path oriented technologies such as MPLS have made constraint-based routing feasible and attractive in public IP networks.

The growth of the Internet leads to more complexity in the routing process; while the emergence of new applications also leads to new service requirements. In order to support the new demands, we believe that the research on routing will be focused on these areas:

- **Inter-domain routing**. As the Internet continues to grow, the number of ASs and the sizes of individual ASs will increase. BGP will probably continue to evolve and new parameters will be defined. Consequently, the complexity of inter-domain routing will also increase. A new routing technology that gives better support to inter-domain routing will be needed in the near future.

- **Multicast routing**. The importance of multicast routing is recognised. Examples of services that require multicast are conferencing, streaming audio and video, and interactive gaming. These applications and services cannot scale to thousands or millions of receivers with multiple point-to-point, unicast streams.

- **Support for mobility**. In the long run, it may well be the case that all computers are mobile. Even so, they will have to be connected to the Internet. Therefore, the Internet Protocol must be extended to support mobility.

- **Supporting QoS requirements**. New applications, such as multimedia or real-time data, are sensitive to delay, and even more sensitive to delay variations (jitter). There are still many challenges concerning how to solve the routing problem in a network where traffic with QoS requirements coexists with best-effort traffic. We will have to offer QoS that satisfies the user requirements while trying to optimise the utilisation of network resources.

- **Support for IP over Optical Networks**. Due to the high bandwidth requirements between the IP routers in the future Internet, we expect that the communication between IP routers in the core network will be implemented by a supporting connection oriented circuit switched network. The strongest candidates are Optical Transport Network (OTN) and SDH based on Packet over SDH (PoS). Thus the IP backbone network will consist of two networks, the connection-less IP network and a connection oriented network. Currently there are heavy research activities on the co-operation between the IP network and the OTN/PoS (by IETF, EURESCOM, etc.).

References

[BELL57] Bellman, R E. 1957. *Dynamic Programming.* Princeton University Press.

[BLAC00] Black, U. 2000. *IP Routing Protocols.* Upper Saddle River, NJ, Prentice-Hall.

[CHEN98] Chen, S, Nahrstedt, K. 1998. An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions. *IEEE Network,* 12 (6), 64–79.

[CR-LDP] *Constraint-based LSP setup using LDP.* Work in progress, Internet Draft <draft-ietf-mpls-cr-ldp-05.txt>, February 2001.

[CR-notes] *Notes on Path Computation in Constraint-Based Routing.* Work in progress, Internet Draft <draft-kompella-te-pathcomp-00.txt>, July 2000 (obsolete document).

[DIJK59] Dijkstra, E W. 1959. A Note on Two Problems in Connection with Graphs. *Numerical Mathematics.*

[EURP616] Eurescom. *State-of-the-art of Routing Principles.* Eurescom P616, Task 2, PIR 2.3.

[Fedyk] *Multiple metrics for traffic engineering with IS-IS and OSPF.* Work in progress, Internet Draft <draft-fedyk-isis-ospf-te-metrics-01.txt>, November 2000.

[FF62] Ford, L R, Fulkerson, D R. 1962. *Flows in Network.* Princeton, New Jersey, Princeton University Press.

[HUIT00] Huitema, C. 2000. *Routing in the Internet.* 2nd Edition. Upper Saddle River, NJ, Prentice Hall.

[ISIS-TE] *IS-IS extensions for Traffic Engineering.* Work in progress, Internet Draft <draft-ietf-isis-traffic-04.txt>, August 2001.

[JUNOS] JUNOS Internet Software Configuration Guide, MPLS applications, release 4.2. *Juniper Networks,* September 2000.

[KESH97] Keshav, S. 1997. *An Engineering Approach to Computer Networking: ATM Networks, the Internet and the Telephone Network.* Reading, Mass., Addison-Wesley.

[MEEL90] Maxemchuk, N, El Zarki, M. 1990. Routing and Flow Control in High-Speed Wide-Area Networks. *Proceedings of IEEE,* 78 (1), 204–221.

[ORDA00] Orda, A, Sprintson, A. QoS Routing: The Precomputation Perspective. In: *Proceedings: IEEE INFOCOM 2000*. Piscataway, NJ, IEEE, 128–136.

[OSPF-TE] *Traffic Engineering Extensions to OSPF.* Work in progress, Internet Draft <draft-katz-yeung-ospf-traffic-05.txt>, June 2001

[QoSGlos] Johnson, V. *Quality of Service – Glossary of Terms.* www.stardust.com, May 1999.

[RFC1771] IETF. Rekhter, Y, Li, T. 1995. *A Border Gateway Protocol-4.* (RFC 1771)

[RFC2002] IETF. Perkins, C (ed.). 1996. *IP Mobility Support.* (RFC 2002)

[RFC2386] IETF. Crawley, E et al. 1998. *A framework for QoS-based Routing in the Internet.* (RFC 2386)

[RFC2439] IETF. Villamizar, C, Chandra, R, Govindan, R. 1998. *BGP route flap damping.* (RFC 2439)

[RFC2702] IETF. 1999. *Requirements for Traffic Engineering over MPLS.* (RFC 2702)

[RSVP-TE] *Extensions to RSVP for LSP tunnels.* Work in progress, Internet Draft <draft-ietf-mpls-rsvp-lsp-tunnel-09.txt>, August 2001.

[STEE95] Steenstrup, M (ed.). 1995. *Routing in Communications Networks.* Englewood Cliffs, NJ, Prentice-Hall.

[SU00] Xun Su, de Veciana, G. 2000. Source Routing in Networks with Uncertainty: Interference, Sensitivity and Path Caching. In: *Proceedings of IEEE GLOBECOM 2000.* Piscataway, NJ, IEEE, 460–464.

[THOM98] Thomas, Thomas, M. 1998. *OSPF Network Design Solutions.* Indianapolis, IN, Macmillan Technical Publishing.

Routing Strategies for IP Networks

WALID BEN-AMEUR, NICOLAS MICHEL, BERNARD LIAU AND ERIC GOURDIN

Walid Ben-Ameur (28) is currently Associate Professor in the INT, National Institute of Telecommunications, Evry, France. Prior to that he was a researcher at France Telecom R&D from 1999 to 2001. He earned his PhD degree in computer science with best honours from the ENST, "Ecole Nationale Supérieure des Télécommunications", in 2000. His research interests span various aspects of network design including graph problems and optimization algorithms. Dr. Ben-Ameur is the author of more than 20 conference and journal papers.

walid.benameur@int-evry.fr

Nicolas Michel (29) was a student of the Ecole Polytechnique from 1991 to 1994 and graduated from ENST (Ecole Nationale Supérieure des Télécommunications) in 1996. He joined France Telecom R&D (former CNET) in 1996 in the "Optimization, Architecture and Traffic" R&D laboratory. His main areas of research have been in network dimensioning and provisioning and in network cost modelling for strategic analysis. Since 1998 he has studied Next Generation Network (NGN) architectures and routing methods for broadband networks (IP, ATM, MPLS). He is now in charge of a project on Traffic Engineering solutions for France Telecom's IP networks.

nicolas.michel@francetelecom.com

This work addresses the problem of static routing complexity and performance for best effort traffic in a data network and more specifically an Internet network running an IGP (*Interior Gateway protocol*), and MPLS if necessary. We first give a short presentation of the various routing strategies (single-path and multi-path) and their possible realisation in an IP intra-domain network. We then briefly introduce the problem of the performance measurement of a routing pattern. We also define the complexity of a routing pattern as the number of MPLS tunnels needed for its realisation. We show how the number of MPLS tunnels that are needed to enhance an IGP routing strategy can be minimised. We compare different routing strategies in IP networks from the two points of view: complexity and performance. We then propose two off-line Traffic Engineering methodologies for IP intra-domain network: the first one is based on an IGP/MPLS architecture; the second one is based only on the IGP routing using an optimised load balancing scheme. The algorithms used to compute the IGP metric and to optimise the routing patterns are also briefly described.

1 Introduction

Traffic routing within a telecommunication network defines how the traffic matrix is mapped on the network topology. Routing mechanisms are thus identified as an essential feature in the control of the network performance [Awduche_1]. The routing mechanisms involved allow assigning the network capacities, more or less efficiently, to the demands. The routing choice has a direct impact on the existence and location of congestion within the network. A high level of congestion may decrease the grade of service (call blocking, increased delays, packet losses, etc.).

Routing mechanisms within an IP network may induce some restrictions on the path choice related to the path selection algorithm. The problem occurs more specifically in the case of IP networks running an IGP (Interior Gateway Protocol) routing protocol. In this case, the routes derive from very simple routing algorithms (shortest path calculations) which offer only limited control over the routing paths. This often leads to a sub-optimal utilisation of the network resources. Today several new mechanisms are proposed to increase the routing control and to optimise the network performance, and among them MPLS. However, such mechanisms also introduce some complexity in the network management. We try to analyse the compromise between routing performance and complexity. We propose two off-line Traffic Engineering methodologies: the first one is based on an IGP/MPLS architecture; the second one is based only on the IGP routing using an optimised load balancing scheme.

2 Organisation of the Paper

We introduce various (static) routing strategies (single-path and multi-path routing strategies) and describe how they can be specifically realised in an IP intra-domain network (Section 3).

We then present some of the routing performance criteria that can be optimised (Section 4). We also introduce the complexity of an IP routing strategy as the number of MPLS tunnels needed.

The performance and complexity of various IP routing strategies are then compared according to the most heavily loaded link criterion (Section 5).

Some classes of efficient routing strategies are selected from these comparisons and two off-line Traffic Engineering methodologies are derived (Section 6).

Section 7 is devoted to the algorithms used in the context of performance optimisation.

3 Some Static Routing Patterns

We first need the following definitions:

Network topology: We assume that we can represent the network topology as a simple non oriented graph that is represented by its nodes and edges. Multiple parallel links are represented by a unique edge between the nodes.

- Note that in MPLS Traffic Engineering although *n* parallel links can be announced as a single bundled link [Kompella], in order to use all links capacity, *n* parallel LSPs must be established (unless a solution based on LSP hierarchy is used [Kompella_2]). For IGP routing see ECMP below.

Routing pattern: For a given network topology, we define a routing pattern as a set of (possibly multiple) *directed* routes between pairs of nodes in the network. If there is at least one route in each direction between each pair of nodes, the routing pattern is fully meshed.

Various static routing patterns are introduced here with their possible realisation in an IP intradomain network. We also focus on some specific IP routing strategies based on the modification of the IGP routing with ER-LSP (Explicit Routed Label Switched Path) created with MPLS.

In the sequel the terms ER-LSP, tunnel, and MPLS tunnel are indifferently used.

3.1 Single-path Routing Patterns

In a single-path routing pattern there is at most one route between each pair of nodes. We can distinguish symmetric single-path routing patterns if the paths between *A* and *B* and *B* and *A* use the same edges for all pairs of nodes (A,B). Single-path routing patterns may be divided into the following interesting sub-classes:

- **Shortest path routings patterns**: If there exists a metric (a set of pairs of values, one for each direction, on the edges of the network) such that all paths of the routing pattern are a shortest path between the end-points according to that metric. A special case is when all shortest paths are also unique (unique shortest path).

  - Classical intra-domain routing protocols (OSPF, IS-IS) are based on such shortest path calculations. Administrative metric values are related to the system interfaces: between two routers a different metric value can be related to each interface of a same link. Resulting routing patterns can thus be symmetric or not.

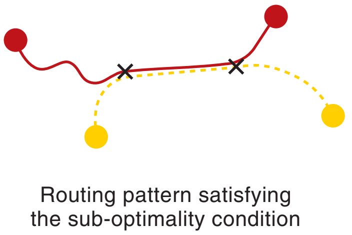
- **Routing patterns satisfying a sub-optimality (SO) property**: Two given paths having two points in common satisfy the sub-optimality condition if they share the same sub-path between these two points (Figure 1). Note that this sub-optimality condition excludes traffic load balancing and load distribution which aim to divide at an intermediate node the traffic toward the same destination on several distinct paths. Note also that routing patterns satisfying the SO condition are necessarily symmetric. Routing patterns based on unique shortest paths satisfy the sub-optimality condition when the metric values are the same on the two interfaces of a link. The contrary is false [Ben-Ameur&Gourdin_1].

- **Destination-based single-path routing**: Any packet is forwarded through the network using the destination address. Obviously, shortest path routing and sub-optimal routing are also based on destination. However, this class of routing patterns is larger. In fact, this is equivalent to establishing a spanning tree for each destination. The destination trees can be completely independent.

- **General single-path routing patterns without constraints**: The whole traffic demand between an origin-destination pair is routed through a single path without any additional constraint.

  - In an IP network running a classical IGP routing protocol, only shortest path routing patterns can be realised. Other single-path routing patterns can be realised with the explicit routing functionality enabled by MPLS (strict ER-LSP). As an ER-LSP is always unidirectional, symmetric or directional routing patterns can be realised. When the routing pattern is fully meshed, the total number of ER-LSP to create is equal to $n * (n - 1)$ where n is the number of nodes.

In the sequel, for the sake of simplicity of the study, we focus our attention on symmetric single-path routing patterns only. Note that for operational reasons this property is often required by network operators. One reason is to limit the complexity of management of the network. Another reason is to prevent having a


Routing pattern satisfying the sub-optimality condition

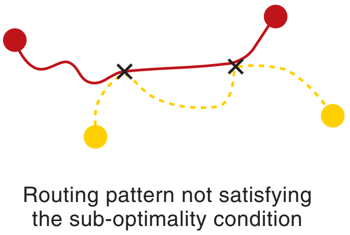

Routing pattern not satisfying the sub-optimality condition

Figure 1 The sub-optimality condition

Bernard Liau (45) graduated from the Ecole Nationale Supérieure des Télécommunications, Paris, France in 1979 and joined France Telecom R&D in 1985. His main research interests have included traffic and network modelling, optimisation of telecom networks and cost oriented pricing. He currently leads the group of France Telecom R&D in charge of multimedia network architecture and optimisation.

bernard.liau@francetelecom.com

Eric Gourdin (38) obtained his PhD in 1994 from the Ecole Polytechnique de Montréal, doing most of his PhD research at the Groupe d'Étude et de Recherche en Analyse des Décisions on global optimization problems. He worked on a grant from 1996 to 1998 at Université Libre de Bruxelles on traffic management problems. Since 1998 he has been working with France Telecom R&D on various network opitmization problems, especially the ones arising in the context of new Internet routing protocols. As of this year he is in charge of a project aimed at developing new models and methods for network optimization problems.

eric.gourdin@francetelecom.com

routing path up in one direction while the return routing path is down due to a link failure. With symmetric routing patterns, routing paths in both directions are simultaneously up or down in case of link failure.

3.2 Multi-path Routing Patterns

In a multi-path routing pattern, traffic between two nodes can be forwarded among several distinct paths.

In IP networks, load sharing can be achieved at an intermediate node in multiple ways: on a packet per packet basis, or with a hashing function evaluated from the information read in the packet header, etc. A hashing function based on the origin and destination can achieve sufficient granularity in a core network.

- An IGP routing protocol can provide multiple equal cost paths between which load sharing can be implemented. Because there is no information in current IGP routing protocols about traffic loading on distant links, techniques have been utilised to divide traffic somewhat evenly among the available paths. Those techniques are referred to as Equal Cost MultiPath (ECMP). A classical utilisation of ECMP is to assign the same metric to parallel links between two routers so that all those links will be used to forward traffic. This is thus equivalent to single-path routing in our topology model where we consider multiple parallel links as a unique (aggregated) link. Another technique, Optimised MultiPath (OMP) [OSPF-OMP], tries to adjust the load balancing parameters at each node in function of the network load. This requires significant changes to the IGP because dynamic information is needed in each router about link loads in the network. This proposition was never implemented;

- General ECMP: Instead of splitting the traffic evenly between the shortest paths, we can split it in any arbitrary way. In fact, it is very easy to see that when no particular routing constraints are added (number of hops for example), the link loads of any multi-path routing pattern can be reproduced by a routing strategy where forwarding is based only on destination. That is to say, a node *B* which has to route a packet to *A*, will randomly choose a path (an interface) using only the destination address. In other terms, if a certain proportion of the traffic demands from *C* to *A* and from *D* to *A*, uses *B* as an intermediate node, then this traffic will be split in the same way between *B* and *A* whatever the origin (*C* or *D*) (Figure 2). We will show in Section 7 how a multi-path routing can be transformed into a shortest path routing;

- With MPLS, several tunnels can be opened between a pair of nodes, and traffic can be arbitrarily shared among them.

3.3 Specific Routing Patterns in IP Networks

The realisation of the routing patterns mentioned above is based either on the IGP routing or on administratively configured TE tunnels. Both mechanisms can be integrated: the IGP routing can be modified to take into account TE tunnels. Three different models can be identified: in the first two models, only the path selection process of the IGP in a node is modified taking into account the TE tunnels originating at this node, in the third model TE tunnels are advertised by the IGP protocol.

- “Basic IGP Shortcut”: If a packet arrives in a router where a tunnel originates with remote egress equal to the destination of the packet, then the packet is forwarded to the destination. Otherwise the packet follows the classical IGP routing;

- “IGP Shortcut”: In this model proposed in the IETF [Smit], the shortest path calculation in the routers remains unchanged but the determination of the next hop is modified in the following way: if a tunnel originates in the router with its egress belonging to the shortest

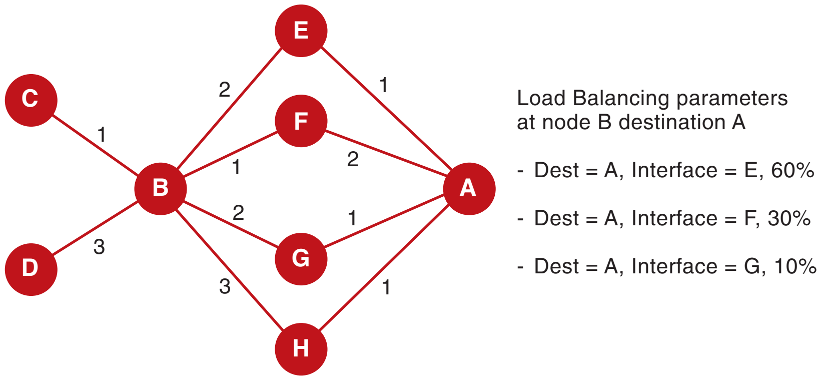

E
C
F
B
A
D
G
H
1
2
1
1
2
2
1
3
3
1
Load Balancing parameters
at node B destination A
- Dest = A, Interface = E, 60%
- Dest = A, Interface = F, 30%
- Dest = A, Interface = G, 10%

Figure 2 General ECMP

path, then the packet will be forwarded in this tunnel;

- “Advertise tunnels into the IGP”: In this model implemented by some manufacturers, tunnels are advertised in the IGP and used in the shortest path calculations as virtual interfaces.

Depending on implementation details and in particular on the tunnels metric assignment, many different options are possible in the path selection process. They give more flexibility to the current IGP routing protocols: the resulting routing patterns will not necessarily be shortest paths, nor satisfy the SO condition, nor even be destination based.

4 Routing Performance Criteria for Best Effort IP Traffic

We consider static routing patterns and best effort traffic controlled by TCP. The performance of routing patterns can be viewed from the user’s point of view or from the network’s point of view. This distinction is introduced in [Awduche_2] where *traffic oriented performance* and *resource oriented performance objectives* are defined:

- Traffic oriented performance: The quality of service perceived by end users is mainly determined by the (random) duration of a document transfer (Web page, e-mail, FTP file, etc.). Since the source traffic rates are reactive to the network load (TCP behaviour), the quality of service will depend on the link loads across the path;

- Resource oriented performance: From the operator’s point of view, the objective is to minimise resource utilisation (link capacity). Another objective can be the robustness of the traffic repartition against traffic fluctuations. The first objective implies that a routing pattern must be found such that another routing cannot be found with a lower load on each link and with a strictly lower load for at least one link. Such a routing pattern is said to be non dominated. The second objective can be partially addressed by looking for a routing pattern that minimises the maximum link load: such a routing pattern will be able to cope with the maximal traffic increase (with the assumption of a homogeneous traffic increase across all origin-destination demands).

For the sake of computational tractability, a simple performance criterion is required: it should only be related to the edge loads and capacities, but independent of the network topology and of the effectively used routing paths.

Notations:
We consider a network defined by its set of edges L and a given static routing pattern. Let C_l be the capacity of edge l and A_l be the average traffic load carried through this edge (this load effectively depends on the routes within the network). The average load of edge l is defined as $\rho_l = A_l / C_l$. A routing pattern is said to be feasible if $\rho_l \leq 1$ for every edge.

Criteria based on the edge loads:
It seems natural to try to maximise a concave decreasing function of the edge loads as for instance:

$$\frac{1}{1-\alpha}\sum_{l\in L}\left(1-\rho_l\right)^{1-\alpha}, \ \alpha \geq 0, \ \alpha \neq 1 \qquad (1)$$

This function was proposed and studied in [Mo&Warland] and [Bonald&Massoulié].

When α is close to 1, the function (1) is equivalent to $\frac{\|L\|}{1-\alpha} + \sum_{l\in L}\log\left(1-\rho_l\right)$.

Therefore, for $\alpha = 1$, criterion (1) can be extended and replaced by $\sum\log\left(1-\rho_l\right)$.

A routing is said to be optimal if it is able to carry the whole traffic flow minimising criterion (1). An interpretation can be proposed for some values of α:

- $\alpha = 0$ minimises the average edge load. This is a simple criterion but we would not recommend it because it is unable to differentiate two links with respective loads of 0 % and 100 % and two links 50 % loaded (contrarily to the case $\alpha > 0$, the function is not strictly concave);
- $\alpha = 1$ maximises $\sum\log\left(1-\rho_l\right)$, equivalently the geometric mean of $(1 - \rho_l)$;
- $\alpha = 2$ minimises $\sum 1/\left(1-\rho_l\right)$, equivalently the harmonic mean of $(1 - \rho_l)$;
- $\alpha = \infty$ corresponds to a “min-max” criterion. One is successively interested in minimising first the maximum load, then the second maximum load, and so on.

The higher the value of α is, the more attention is paid to the most heavily loaded edge.

Criteria based on the edge residual capacities:
It is also possible to replace in (1) the edge load by the residual capacity $C_l(1 - \rho_l)$. Objective

functions of the following type can thus be considered:

$$\frac{1}{1-\alpha}\sum_{l\in L}\left(C_l(1-\rho_l)\right)^{1-\alpha}, \quad \alpha \geq 0,\ \alpha \neq 1 \quad (2)$$

Interpretations similar as for criterion (1) can be proposed. The higher the value of α, the more attention is paid to the edge with the lowest residual capacity.

Note that a routing pattern achieving the optimal value for one of the criteria described above is a non-dominated solution.

The choice of a performance objective can be driven by the nature of the studied network, backbone or access network. Considering a backbone network, the customer bit rate is generally bounded by the access rate (or the rate of the Web server) which is small compared to the edge capacities. The *traffic oriented performance* criteria are thus less crucial than the *network oriented performance* ones. A criterion related to the most heavily loaded edge seems relevant in the case of static routing when the network is unable to automatically adapt to traffic fluctuations. The most heavily loaded edge criterion is one of the most often used criteria to evaluate the performance of backbone networks.

5 Comparison of Static Routing Patterns

The following static routing strategies are compared (listed in a decreasing order of flexibility):

- Multi-path symmetric routing;
- Single-path symmetric routing;
- Single-path symmetric routing with constraint of sub-optimality;
- Unique symmetric shortest path routing;
- Minimum hop (symmetric) routing.

In the sequel, it is implicit that all routing patterns considered are symmetric. We believe some of the results can be extended to asymmetric routing patterns but this is left for further study.

Bear in mind that for any multi-path routing pattern, it is possible to find a destination based multi-path routing scheme that achieves the same load links (see Section 3.2). This routing scheme can be implemented using a generalised ECMP technique.

Definitions:

1) For a given routing strategy and a given network topology, we call *routing set* of a routing strategy the set of all routing patterns that can be achieved with this routing strategy;

2) For a given routing strategy, a given network topology, and a given performance criterion, we call *performance* of a routing strategy the best performance of all routing patterns that can be achieved with this routing strategy.

We first define the notion of complexity of a routing strategy in an IP network. We then try to analyze the various routing patterns that can be achieved with the above routing strategies and the associated complexity. Finally we compare the performance of these routing strategies.

5.1 Complexity of the Realisation of a Routing Pattern in IP Networks

The IGP routing protocol has some advantages: its simplicity, scalability, automated and distributed implementation. Moreover IGP routing has already proven its robustness and resilience. A disadvantage of using MPLS explicit routes is the administrative burden and potential for human induced errors from using this approach on a large scale [Michel&al]. Network operators might thus want to minimise the total number of MPLS tunnels created in the network.

Definition:
We define the complexity of a routing pattern as the number of tunnels that are needed for its realisation in an IP network.

5.2 Scenarios

Several scenarios (topology and traffic matrix) have been selected in order to compare the different routing strategies. Some of them have been studied by C. Villamizar [Villamizar_1, Villamizar_2] in the evaluation of OMP approaches and the others have been extracted from real case world networks.

The scenarios used by Curtis Villamizar are available on his Web site along with the results of his simulations [Villamizar_2].

These scenarios are defined by a network topology (obtained by random generation) along with capacity on the edges and a traffic matrix. Edges

	Nodes	Edges	Mesh degree	Demands
OMP_10_29	10	29	5.8	45
OMP_20_51	20	51	5.1	190
OMP_50_101	50	101	4.0	1225

Table 1 Results of compatibility for various routing patterns

	Number of compatible routing patterns		Percentage of compatible paths (in case of non compatible routing pattern)	
	General single-path routing pattern	Sub-optimality compliant routing pattern	General single-path routing pattern	Sub-optimality compliant routing pattern
OMP_10_29	0 %	51 %	35 %	95 %
OMP_20_51	0 %	2 %	29 %	88 %
OMP_50_101	0 %	0 %	33 %	69 %

are symmetric but may have a different capacity in each direction. The traffic matrix is oriented.

The two following scenarios extracted from real case networks have also been studied:

- Scenario FT_1: 9 nodes 20 edges and 35 symmetric demands;

- Scenario FT_2: 26 nodes 39 edges and 154 symmetric demands.

5.3 Comparison of Routing Sets: Size and Complexity

In what follows, we try to answer the following questions: what is the relative size of the routing sets of each routing strategy? What is the complexity of realisation of the corresponding routing patterns in an IP network?

5.3.1 Shortest Path Routing

We first introduce some definitions:

1) *A single path and a metric are compatible* if the path is a unique shortest path according to the metric. A metric is compatible with a single-path routing pattern if all paths are compatible with the metric. In Section 7, we address the case where the constraint of uniqueness of a shortest path is relaxed;

2) *A routing pattern is compatible* if there exists a metric compatible with all paths in the routing pattern;

3) For a given single-path routing pattern *the number of compatible paths* is defined as the maximal number of paths of a compatible sub-routing pattern (a subset of paths of the routing pattern).

A first step in this routing strategy analysis is to measure the difficulty to find compatible metrics for a given routing pattern. For different network topologies, we have randomly generated 100 fully meshed single-path routing patterns and 100 fully meshed single-path routing patterns satisfying the sub-optimality condition. In each case a compatible metric has been searched using a linear programming method described in [Ben-Ameur&Gourdin_1] and [Ben-Ameur& Liau] (see Section7). Results are displayed in Table 1.

Bear in mind that a routing pattern that is not satisfying the sub-optimality condition is never compatible [Ben-Ameur&Gourdin_1].

Although a limited number of topologies has been tested, we can draw the following trends from these results:

- *General single-path routing patterns:* It seems difficult to find a compatible metric for general single-path routing patterns (not a single case in our tests). The routing set of single-path routing strategy is thus much larger than the routing set of the unique shortest path routing strategy. However it is possible to find a metric compatible with at least a significant sub-routing pattern: in average 30 % of the paths whatever the size of the network;

- *Sub-optimality compliant routing patterns:* In a significant number of cases it is possible to find a compatible metric. The size of the routing set of the sub-optimality compliant routing strategy seems to be very close to the size of the routing set of the unique shortest path routing strategy for (very) small networks (scenario OMP_10_29). As the size of the network increases (a few dozen nodes), the size of the routing set of the sub-optimality compliant routing strategy seems again to be much bigger than the size of the routing set of the unique shortest path routing strategy (scenario OMP_20_51 and OMP_50_101). However the percentage of compatible routing paths is higher than for the general routing patterns (more than 70 %) although it seems to decrease with the size of the network.

These results depend on the studied topologies. For example, for a ring network the routing set

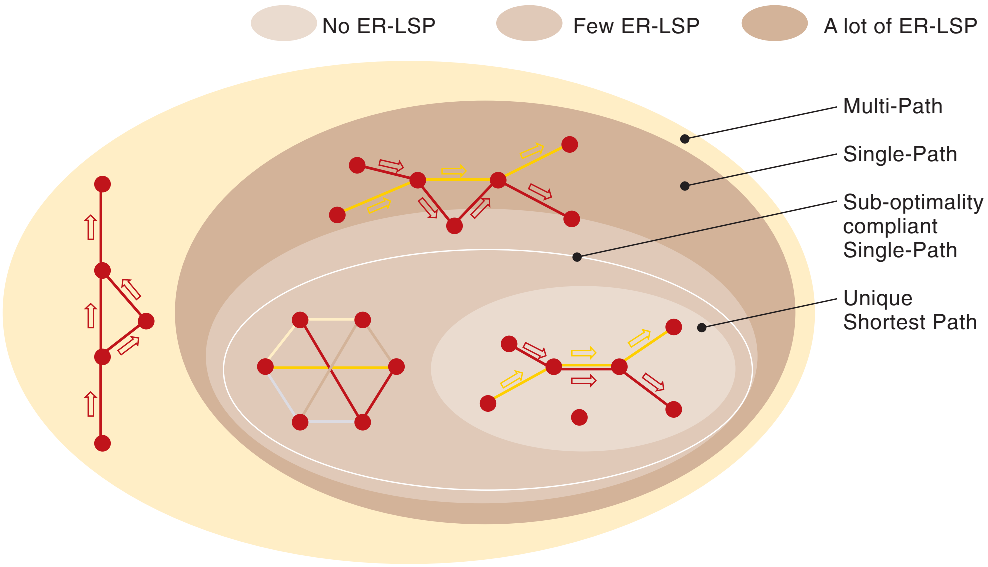


Figure 3 Complexity of various routing patterns

of the sub-optimality compliant routing strategy is equal to the routing set of the unique shortest path routing strategy [Ben-Ameur&Gourdin_1]. It is likely that the results depend on the degree of connectivity of the network. Other relevant topologies for IP networks are under study.

5.3.2 Single-path Routing with Metrics and Tunnels

We have seen that a general single-path routing pattern is often not compatible. It is possible to realise such routing patterns in an IP network using strict explicit routing, for example by creating two ER-LPS per path, one in each direction. This requires $n * (n - 1)$ MPLS tunnels in the network (if the routing pattern is fully meshed). The routing complexity is thus directly related to the number of demands.

However in the case of sub-optimality compliant routing patterns, it is often possible to find a metric compatible with a large percentage of the paths in the routing pattern. The question is now the following: is it possible to reproduce the remaining non-compatible paths with the IGP routing modified with a limited number of MPLS tunnels?

We consider the "IGP Shortcut" model of integration of the IGP routing with the MPLS tunnels (Section 3.3). For each remaining path not compatible with the metric, the two corresponding ER-LSP are created (one in each direction). The modified IGP routing will thus route the traffic along the correct paths for these routing paths not compatible with the metric. However those tunnels can modify the routes found by the modified IGP for the paths that are compatible with the metric.

It is easy to show the following result: if the initial routing pattern satisfies the sub-optimality condition, then the tunnels created as described above do not modify the IGP routing for the paths that were compatible with the metric. Thus, in the case where the routing pattern satisfies the sub-optimality condition, it can be realized by an IGP routing protocol modified by some tunnels. The number of pairs of tunnels (one in each direction) needed is equal to the number of paths in the routing pattern minus the number of compatible paths. However, in some cases, it may be possible to create less tunnels because a pair of tunnels may modify more than one shortest path into the correct routing path (see Section 7.1.2).

5.3.3 Complexity of the Routing Patterns

We consider all routing patterns (including single-path and multi-path routing patterns) and their realisation in IP networks. Some of them can be reproduced without any MPLS tunnels (i.e. using only the IGP routing), some others require the creation of a limited number of MPLS tunnels (IGP routing modified with some MPLS tunnels) and the last routing patterns require a large number of MPLS tunnels (in the order of the number of paths in the routing pattern).

Based on the results above, we can represent in Figure 3 a comparison of the complexity of different routing patterns.

We can see that a large number of routing patterns (much larger than the number of routing patterns that can be achieved with the IGP routing only) can be achieved with a "reasonable" complexity (with a limited number of tunnels). The natural question that arises is the following: what level of performance can be achieved with each level of complexity?

Results	**Multi-path**	**Single-path**	**Minimum Hop Routing**	**Unique Shortest Path**
OMP_10_29	**0.61 (MPLS-OMP)**	**0.83**	1.15	0.85
OMP_20_51	**0.70 (MPLS-OMP)**	**1**	1.82	0.87
OMP_50_101	**0.69 (MPLS-OMP)**	**0.88**	1.60	0.82
FT_9	0.78*	0.79*	2.93	0.80
FT_26	0.64*	0.66*	1.50	0.88

Table 2 Performance of different routing strategies

5.4 Comparison of Performance

The performance criteria considered in this Section concern the network's ability to support traffic increases. It is measured by the maximum edge load (Section 4).

5.4.1 Optimisation

A different optimisation problem has to be solved for each routing strategy. Some of them are NP-hard and cannot be solved exactly: in these cases a heuristic has been used. As a consequence, the comparison of the routing strategy performance may be affected by the accuracy of these heuristics. The routing optimisation procedures we have used are described below:

- *Multi-path routings:* A linear programming (exact solution);
- *Single-path routings:* A heuristic (a branch and cut algorithm) based on linear programming which also provides an upper bound on the optimal solution [Geffard]. Only symmetric problems can be solved with this tool (consequently not the Villamizar scenarios[1]);
- *Single-path with constraint of sub-optimality:* An exact solution (based on a linear programming) is under study [Ben-Ameur&Gourdin_2].
- *Unique shortest path:* A simulated annealing heuristic [Ben-Ameur&al].

More details about these optimization algorithms are given in Section 7.

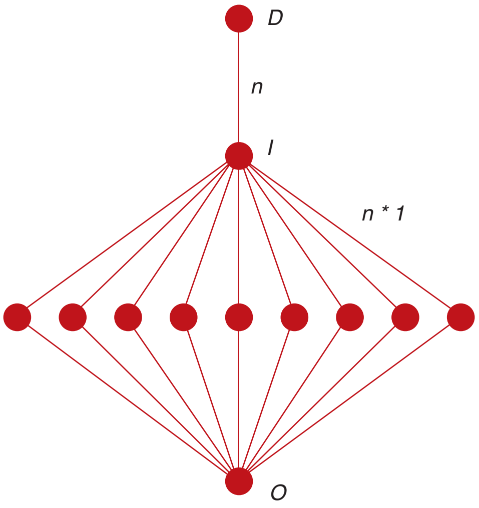
5.4.2 Results

Table 2 summarises the main results of our tests. In order to understand this table, note that:

- A result marked with * means that the solution value is optimal;
- Results in bold characters were obtained by Villamizar and are directly reported from his Web site [Villamizar_2]: results for MPLS-OMP are used for the multi-path routing strategy and the single-path routing strategy (results are obtained with a simple greedy heuristic).

The following comments can be derived:

- *Single-path versus multi-path routing:* In the case of scenarios FT_9 and FT_26, the proposed solution is optimal and the performance of both routing strategies is very close. The result is quite different in the case of Villamizar scenarios. The single path constraint decreases the performance (about 30 %). Note that in the latter case the optimisation heuristic used is very simple and we have no guarantee of the quality of the solution. Results seem to depend highly on the network topology and on the traffic matrix. Note that it is easy to build scenarios for which the performance of the single-path routing strategy is arbitrarily worse than the performance of the multi-path routing strategy (below is an example of a topology on which a single-path routing strategy will perform very badly compared to a multi-path routing strategy because it is not possible to balance the traffic from *O* to *D* on the *n* parallel paths). However, in an operational perspective, the worst case is not relevant, only the average case over realistic topologies;



- *Shortest path routing versus minimum hop routing:* The comparison between unique shortest path routing and minimum hop routing strategies illustrates the significant impact of a wise selection of the metric values. The choice of a default value (in the minimum hop

[1] *Results for the Villamizar scenarios are directly reported from his Web site [Villamizar_2].*

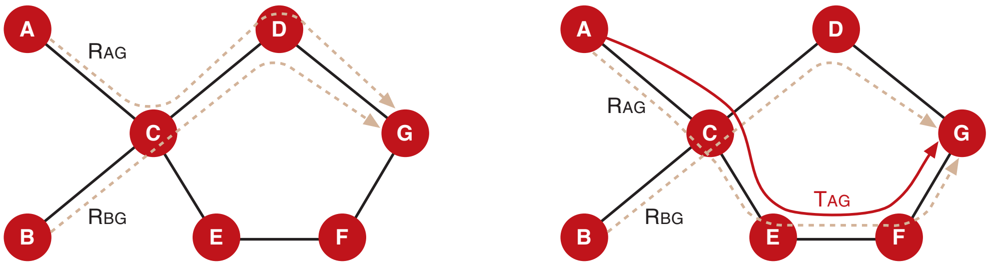


Figure 4 Shortest path routing pattern modified by a TE tunnel thereby achieving load balancing

routing strategy, the edge metric value is systematically set to one) may induce a very poor performance compared to the performance achievable with an optimised metric (in the studied scenarios, the relative performance drops from 25 % up to 200 %);

- *Single-path routing versus unique shortest path routing:*
  - Note that for the Villamizar scenarios, the performance achieved with unique shortest path strategy is sometimes better than with a less constrained single-path routing strategy. It only means that, in the case of single-path routing optimisation, the heuristic is not accurate enough to reach a value close to the optimum. This may be of some importance, because such heuristics are quite often used, even in operational network configuration tools;
  - In the case of FT_9 and FT_26 scenarios, the optimal performance of the single-path routing strategy is found. For the smaller network (FT_9), the performance that can be achieved with the unique shortest path strategy is very close to this value. However for scenario FT_26, the best performance that can be achieved with the unique shortest path strategy is 30 % worse than this value. Further tests are needed to investigate whether the gap increases with the size of the network (number of edges).

5.4.3 Performance Improvement with MPLS Tunnels

The size of the routing set for the unique shortest path routing strategy modified with a few MPLS tunnels is much larger than the size of the routing set for the unique shortest path routing strategy. A natural question then follows: is it possible to significantly improve the performance of unique shortest path routing by adding a few MPLS tunnels?

We suppose that the IGP routing is modified by the MPLS tunnels according to the "IGP Shortcut" integration model (Section 3.3). For example, if we consider scenario OMP_10_29, the best performance achieved with the unique shortest path routing strategy is 0.85. By looking at the routing paths, we note that 3 links have the maximum load of 85 %. We have identified 3 pairs of MPLS tunnels that lead to a modified routing pattern where the most heavily loaded link has a load of 77 %.

By creating a few MPLS tunnels, it is in some cases possible to realise a new routing pattern with a significantly improved performance. An important point to mention here is that the resulting routing pattern does not necessarily satisfy the sub-optimality condition. This means that it is possible to achieve some kind of load distribution where two demands may be routed on two paths with two nodes in common but using a distinct path between the 2 nodes (Figure 4).

Finally, note that it is not clear which of the three different models of integration of the IGP routing with MPLS tunnels is the most interesting. The first one, however, may add more complexity because one tunnel can be used by only a limited number of demands.

6 "Off-line" Traffic Engineering Methodologies

Based on the results of Section 5, we can propose off-line "Traffic Engineering" methodologies. The objective is to improve the performance of the network in terms of resource utilisation. Two different methodologies are described: the first one using MPLS, the second one relying on the IGP routing only but using a generalised ECMP technique. In both cases, a single class of (best effort) traffic is considered. It is also assumed that a representative end-to-end traffic matrix between the network nodes can be measured or estimated.

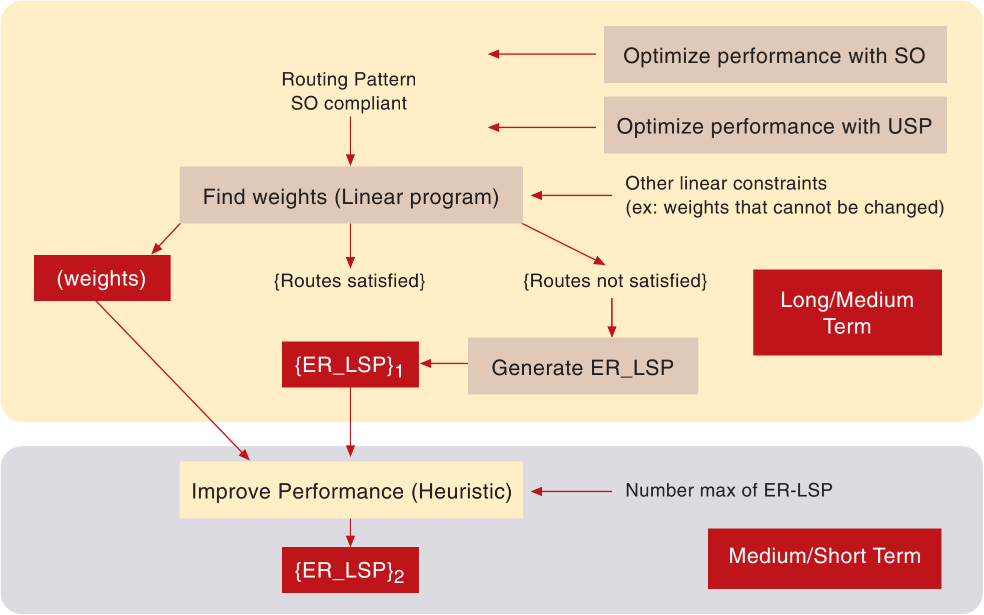
6.1 An MPLS-based off-line Traffic Engineering Methodology

The following assumptions are made:

- MPLS is deployed in the network and it is possible to create explicitly routed MPLS tunnels (ER-LSP);

- The IGP routing is modified to take into account the MPLS tunnels in the determination of the next hop according to the "IGP Shortcut" model (Section 3.3).

Figure 5 Off-line Traffic Engineering methodology


Optimize performance with SO
Optimize performance with USP
Routing Pattern SO compliant
Find weights (Linear program)
Other linear constraints (ex: weights that cannot be changed)
(weights)
{Routes satisfied}
{Routes not satisfied}
Long/Medium Term
{ER_LSP}1
Generate ER_LSP
Improve Performance (Heuristic)
Number max of ER-LSP
{ER_LSP}2
Medium/Short Term

The methodology is depicted in Figure 5. It involves the following steps:

- *Step 1*. First optimise in an off-line procedure the routing pattern according to the performance criteria chosen (for example, try to minimise the load of the heaviest loaded link) allowing either all sub-optimality compliant single-path routing patterns or unique shortest paths routing patterns only. The output is a single-path routing pattern satisfying the sub-optimality condition;

- *Step 2*. Search a metric compatible with a number of paths in this routing pattern equal to the number of compatible paths of the routing pattern. This step can also include some extra constraints provided that they can be expressed using a linear formulation (for example, equalities or inequalities verified by the metric values, minimising the value changes from an existing metric set);

- *Step 3*. If the metric obtained in Step 2 is not compatible with the entire routing pattern obtained in Step 1, create the necessary MPLS tunnels (ER-LSP) in order to reproduce completely the routing pattern obtained in Step 1 (Section 5.3.2);

- *Step 4*. Then try to improve the routing performance of the solution obtained in Step 3 by adding a few MPLS tunnels: it is necessary in this step to find a trade-off between the number of tunnels created and the gain in performance.

We can identify two different parts in this methodology. The first one (Steps 1 through 3) implies the modification of administrative metric values of the IGP in the network. This operation is not desirable to do too often. This type of action can be considered in a medium or long-term basis. The second part of the methodology only attempts to create (or modify) MPLS tunnels in order to improve the routing performance. The tunnel creation and the resulting modification of the routing pattern (calculated by the modified IGP) are simple and fast operations (compared to the IGP convergence). This can be considered as a short-term action.

One of the advantages of this TE methodology is to rely as much as possible on the IGP routing which has already proven its scalability, reliability and which is automated. The administrative metric values are changed when needed in order to optimise the routing performance of the nominal routing pattern. The use of MPLS tunnels enables the network operator to significantly improve the routing performance in response to events in the network (transient change of traffic profile etc.) while limiting the number of MPLS tunnels which limits the complexity of management.

6.2 An ECMP-based off-line Traffic Engineering Methodology

We assume that the routers are able to split the traffic through different equal cost paths (see Section 3.2). The load splitting parameters have to be administratively configured.

The methodology involves the following steps:

- *Step 1*. First compute off-line a multi-path routing pattern optimising the performance criteria chosen (for example, try to minimise the load of the heaviest loaded link). This is generally easy to achieve (see Section 7.2);

- *Step 2*. Determine the destination based multipath routing pattern that achieves the same load links. In other words, determine the adequate load balancing parameters at each intermediate node and for each destination so that the resulting hop-by-hop routing achieves the same link loads (see Section3.2);

- *Step 3*. Compute a metric compatible with this routing pattern (see Section 7.1.3).

We note that with this methodology, both IGP metrics and load balancing parameters must be administratively configured. The operation of modification of administrative metric values of the IGP in the network can be considered on a medium or long-term basis. The operation of modifying load-balancing parameters however does not have any convergence consequence. This could be done on a more frequent basis in response to events in the network (transient change of traffic profile, etc.).

7 Algorithms for Traffic Engineering

In this Section we briefly present some of the algorithms used to address the problems that arise in the context of traffic engineering as described above. Due to space limitation, it is not possible in this paper to give neither the proofs nor the whole details of the algorithms. However, this Section is self-contained and can be understood easily.

7.1 Compatible Metrics

This Section is devoted to methods used to compute a set of edge metrics compatible with a set of routing paths.

7.1.1 Unique Shortest Paths

First let us focus on the case of unique shortest paths. As said in Section 3, the sub-optimality condition (Figure 1) of the routing paths is a necessary condition to find a set of compatible metrics.

Let $G = (V,E)$ be the graph associated with the network. The set of node pairs of G for which a routing path R is given is denoted by K. In other terms, we assume that a path $R(a,b)$ is given for each $(a,b) \in K$. If c and d are such that $c \in R(a,b)$ and $d \in R(c,b)$, then $R(c,d)$ is assumed to be the sub-path of $R(a,b)$ linking c to d (by sub-optimality). $S(a,b)$ is defined as the set of paths between a and b which are different to $R(a,b)$. The metric is denoted by $(m_e)_{e \in E}$.

A general linear model that can be used to find metrics is the following:

$$(LP_1)\begin{cases}\text{Find}(m_e)_{e\in E}\\ \text{Subject to :}\\ \sum_{e\in R(a,b)} m_e = y_{ab};\ \forall (a,b)\in K\\ \sum_{e\in p} m_e \geq 1 + y_{ab};\ \forall (a,b)\in K,\ p\in S(a,b)\\ m_e \geq 0;\ \forall e\in E\end{cases}$$

This linear program can be solved by generalised linear programming. An equivalent polynomial formulation can also be given [Ben-Ameur&Gourdin_1] [Ben-Ameur&Liau]. If a solution is found, the metric given by LP_1 is compatible with the routing paths: every path $R(a,b)$ is a unique shortest path, according to this metric, between a and b.

Note that many particular constraints can be added to LP_1:

- All the metric values must be larger than 1;

- We may also want some links to have equal metrics;

- The routing paths used during failures are also given in advance (they must be shortest paths in the resulting graph obtained after the failure);

- The metrics may be required to be integer.

LP_1 can also be solved considering various kinds of objective functions: minimise the maximum metric, the sum of metrics, or any linear function of the variables, etc.

Note that LP_1 does not always have a solution. Said another way, the sub-optimality condition is a necessary but not always a sufficient condition to find a metric. Some other necessary conditions are proposed in [Ben-Ameur&Gourdin_1]. However, we showed that the sub-optimality is sufficient for some graphs such as cycles, cactus, etc.

In the case where there is no feasible solution, an interesting particular formulation of LP_1 is the one maximising the number of demands whose routing paths are unique shortest paths (or equivalently that maximises the number of compatible paths):

$$
(LP_2)\begin{cases}
\text{Maximize } \sum_{(a,b)\in K} \varepsilon_{ab} \\
\text{Subject to:} \\
\sum_{e\in R(a,b)} m_e = y_{ab};\ \forall (a,b)\in K \\
\sum_{e\in p} m_e \geq \varepsilon_{ab} + y_{ab};\ \forall (a,b)\in K,\ p\in S(a,b) \\
m_e \geq 0;\ \forall e\in E \\
0 \leq \varepsilon_{ab} \leq 1;\ \forall (a,b)\in K
\end{cases}
$$

LP_2 always has a solution. It is also easy to show that the variables ε_{ab}, obtained by solving LP_2, will be equal to 1 or 0. Said another way, LP_2 gives exactly the demands that can be satisfied (in terms of unique shortest path constraint). The objective function of LP_2 can also be more general.

7.1.2 Single-path Routing with Metrics and Tunnels

When a compatible metric cannot be found (because the routing pattern is not compatible or because extra constraints have been added to the linear program), the routing pattern can be reproduced by introducing a few tunnels in order to modify the IGP routing according to the "IGP Shortcut" model (Section 5.3.2). In order to minimise the number of MPLS tunnels that need to be added a linear formulation slightly different from LP_2 can be used. Instead of considering all the LP_2 of $S(a,b)$, we consider only the set $N(a,b)$ of paths that are node disjoint with $R(a,b)$. The program solved is the following.

$$
(MIP_3)\begin{cases}
\text{Minimize the number of tunnels} = \sum_{(a,b)\in K} t_{ab} \\
\text{Subject to:} \\
\sum_{e\in R(a,b)} m_e = y_{ab};\ \forall (a,b)\in K \\
\sum_{e\in p} m_e \geq 1 - \varepsilon_{ab} + y_{ab};\ \forall (a,b)\in K,\ p\in N(a,b) \\
0 \leq m_e \leq M;\ \forall e\in E \\
\varepsilon_{ab} \geq 0;\ \forall (a,b)\in K \\
t_{ab} \geq \dfrac{\varepsilon_{ab}}{1+\left\|R(a,b)\right\|M};\ \forall (a,b)\in K \\
t_{ab} \in \{0,1\};\ \forall (a,b)\in K
\end{cases}
$$

We assume in MIP_3 that the metric values are bounded by a maximum value M. We also use $\|R(a,b)\|$ to denote the number of hops of route $R(a,b)$. The variable t_{ab} indicates whether it is necessary to create a tunnel between a and b. Note that a tunnel is created only if there is a path disjoint with $R(a,b)$ having a cost less or equal to the cost of $R(a,b)$. In the other cases, even if $R(a,b)$ is not a unique shortest path, we do not need a tunnel between a and b because some other intermediate tunnels will be created and used by the demand (a,b) ("IGP Shortcut" model).

MIP_3 can be replaced by other easier linear programs that give a good approximation of the number of tunnels (without the upper bound M):

$$
(LP_4)\begin{cases}
\text{Minimize } \sum_{(a,b)\in K} \varepsilon_{ab} \\
\text{Subject to:} \\
\sum_{e\in R(a,b)} m_e = y_{ab};\ \forall (a,b)\in K \\
\sum_{e\in p} m_e \geq 1 - \varepsilon_{ab} + y_{ab};\ \forall (a,b)\in K,\ p\in N(a,b) \\
0 \leq m_e;\ \forall e\in E \geq 0 \\
0 \leq \varepsilon_{ab} \leq 1;\ \forall (a,b)\in K
\end{cases}
$$

7.1.3 Multi-path Routing Pattern

We assume that a set of paths $R_1(a,b)$, $R_2(a,b)$, ..., is given between each pair of vertices $(a,b) \in K$. We would like to compute a metric such that all these paths are shortest paths. Let $C(a,b)$ be the set of paths between a and b different from the given routing paths $R_1(a,b)$, $R_2(a,b)$, ...

Obviously, a null metric is a solution of the problem. However, for practical reasons, we want to minimise the number of links with a null metric value. This is formulated below:

$$
(LP_5)\begin{cases}
\text{Minimize } \sum_{e\in E} \varepsilon_e \\
\text{Subject to:} \\
\sum_{e\in R_i(a,b)} m_e = y_{ab};\ \forall (a,b)\in K,\ \forall R_i(a,b) \\
\sum_{e\in p} m_e \geq y_{ab};\ \forall (a,b)\in K,\ \forall p\in C(a,b) \\
m_e \geq 1 - \varepsilon_e;\ \forall e\in E \\
0 \leq \varepsilon_e \leq 1;\ \forall e\in E
\end{cases}
$$

The optimal solution of LP_5 is necessarily integer: variables ε_e will be equal to 0 or 1.

Recall that any optimal multi-path routing without particular routing constraints (such as length constraints), can be seen as an optimal routing based only on destination. As LP_5 provides a metric which is compatible with any multi-path routing, we can deduce that it is possible to optimise the network performance only by using a modified ECMP mechanism (Section 3.2). Said another way, first we have to compute an optimal multiflow optimising the performance criterion (for example the maximum load). Then we can determine the load balance coefficients by very simple calculations and transform the multiflow into a multi-path routing based only on destinations. Finally, we compute the edge metrics solving LP_5 (or any other variation of LP_5).

7.2 Optimisation Algorithms

Routing performance optimisation is often a non-trivial problem. Adequate models and methods have to be developed to address each specific problem. Often an exact resolution will not be possible in a reasonable computational time because some problems are NP-hard. In such cases efficient heuristics have to be found. Note that the difficulty of the optimisation problem associated with a given routing strategy can be a decision criterion for an operational application. We present briefly in this Section the different problems and how they can be addressed.

7.2.1 Multi-path Routing Strategies

When multi-path routing is considered, the problem may be easy to solve. For example, if the optimisation criterion is the maximum load or any linear function depending on edge loads, then the problem is polynomial (classical multiflow problem). Moreover, it is easy to integrate some additional constraints. For example one can restrict the problem to paths with limited number hops, etc.

Multiflow problems are very classical. However, some simple and important results are not well known. Suppose for example that we would like to minimise the maximum load. It is very easy to show that we can find an optimal solution such that the number of used paths is lower than the number of demands plus the number of edges. This means that many demands in an optimal solution will be single-path routed.

7.2.2 Single-path Routing Strategies

For general single-path optimisation problems, we use the tool described in [Geffard]. This tool is based on a branch&cut algorithm.

The single-path routing with sub-optimality condition was studied in [Ben-Ameur&Gourdin_2]. The algorithm used to compute a metric satisfying the sub-optimality condition is based on a cutting plane algorithm. To impose the sub-optimality condition, we define two new sets of 0–1 variables: r_e^k and r_v^k for each traffic demand k, each vertex v and each edge e. The sub-optimality condition can be written in the following way:

$$r_e^{a,b} \geq r_c^{a,b} + r_e^{a,c} - 1$$

Many valid inequalities have been introduced to accelerate the algorithm of [Ben-Ameur&Gourdin_2].

Finally, the optimisation problems corresponding to shortest path routing strategies have been solved using some local search algorithms (see [Ben-Ameur&al] and [Michel&al]). The advantages of this method are, first its flexibility: it can be used for different kinds of optimisation criteria and can integrate various constraints related to quality of service. Second it can solve large size problems. The main principle of these algorithms consists in changing the metric of some edges and re-computing the routing paths at each iteration. Some survivability constraints and the multi-hour behaviour of the traffic have been considered in [Ben-Ameur]. Other heuristics have been proposed in [Pioro&al] and [Thorup&al].

8 Conclusion

To summarise, we describe new intra-domain routing mechanisms in IP network and how they can improve routing flexibility and performance in IP networks. Based on some numerical results, we then propose two different off-line Traffic Engineering methodologies that illustrate two possible evolutions of IP routing in intra-domain networks. Necessary algorithms to implement those methodologies are also briefly presented.

A) MPLS-based Traffic Engineering Methodology

A new mechanism like MPLS tunnels explicit routing gives more control over routing in IP networks. Various routing strategies for best effort traffic using this new functionality can be considered and all possible routing patterns can be realised in IP intra-domain network. These routing strategies give more or less flexible control over the routing of the traffic but should also be compared in terms of complexity, scalability and robustness.

The comparison of the performance of these different routing strategies with the criteria of the heaviest loaded link shows that:

- The difference in terms of routing performance of the different routing strategy seems to strongly depend on the size and topology of the studied networks (which is not very surprising). It is thus important to focus on relevant topologies for IP networks;

- Whatever the routing strategy considered, optimisation has an important consequence on the routing performance. This is particularly true for the strategy of unique shortest path routing according to an administrative metric: a wise choice of the metric can significantly improve the routing performance;

- A routing strategy that permits to realise much more various routing patterns cannot necessarily achieve a significantly better performance. A unique shortest path routing strategy performs very well in general and sometimes close to the optimum achievable with single-path or even multi-path routing strategies;

- The use of explicitly routed MPLS tunnels can improve the performance of routing. We show however that it is not necessary to rely only on explicit routing (which requires a large number of tunnels), but that mixed routing strategies based on IGP routing and MPLS tunnels can produce very interesting routing patterns in terms of performance. We give an algorithm minimising the number of MPLS tunnels that need to be added to reproduce a given single-path routing pattern;

Based on those results, an off-line *Traffic Engineering* methodology is proposed. It is based on an optimisation of the IGP routing (by a wise choice of the administrative metrics) enhanced by the use of a limited number of explicitly routed MPLS tunnels. Advantages of such a Traffic Engineering system would be to benefit from the highly proven robustness of the IGP routing while improving the performance and reactivity of the routing control in terms of resource utilisation with a limited added operational complexity.

B) ECMP-based Traffic Engineering Methodology

We assume that routers are able to split the traffic towards one destination on multiple paths according to some administratively defined load balancing parameters. It is then possible to reproduce the same (optimal) link loads in the network as those resulting from any given (optimal) multi-path routing pattern. This does not require any MPLS tunnels.

However, MPLS can integrate various types of routing constraints allowing to implement specific routing strategies and QoS policies.

Acknowledgement

We wish to thank Jérome Geffard for providing his single path optimization software.

9 References

[Awduche_1] Awduche, D et al. *A framework for Internet Traffic Engineering*. (draft-ietf-tewg-framework-05.txt), June 2001.

[Awduche_2] Awduche, D. 1999. MPLS and Traffic Engineering in IP Networks. *IEEE Communications Magazine,* 37 (12), 42–47.

[Ben-Ameur&al] Ben Ameur, W et al. 2000. Designing Internet networks. In: *Proc. DRCN 2000, Reliable Networks for the Information Age,* 56–61, Munich, Herbert Utz Verlag.

[Ben-Ameur&Gourdin_1] Ben Ameur, W, Gourdin, E. Internet routing and related topology issues. Submitted to *SIAM journal of discrete mathematics* (2000).

[Ben-Ameur&Gourdin_2] Ben-Ameur, W, Gourdin, E. *An exact method to optimize IP networks*. 2000. (FT R&D, Internal Technical Report.)

[Ben-Ameur&Liau] Ben-Ameur, W, Liau, B. 2001. Computing Internet routing metrics. *Annals of telecommunications,* 56 (3-4), 150–168.

[Ben-Ameur] Ben-Ameur, W. Multi-hour design of survivable Internet networks. Submitted to *Telecommunications Systems,* 2000.

[Geffard] Geffard, J. 2001. A 0-1 model for singly routed traffic in telecommunication networks. *Annals of Telecommunications,* 56 (3-4), 140–149.

[Kompella] Kompella, K, Rekhter, Y, Berger, L. *Link bundling in MPLS Traffic Engineering*. (draft-kompella-mpls-bundle-04.txt), May 2000.

[Kompella_2] Kompella, K, Rekhter, Y. *LSP hierarchy with MPLS TE*. (draft-kompella-lsp-hierarchy-00.txt), December 2000.

[Li] Li, T. 1999. MPLS and the Evolving Internet Architecture. *IEEE Communications Magazine*, 37 (12), 38–41.

[MATE] Widjaja, I, Elwalid, A. *MATE : MPLS Adaptive Traffic Engineering*. (draft-widjaja-mpls-mate-01.txt), October 1999.

[Michel&al] Ben-Ameur, W et al. Optimizing administrative weights for efficient single-path routing. In: *Networks 2000.*

[Mo&Walrand] Mo, J, Walrand, J. 2000. Fair End-to-End Window-Based Congestion Control. *IEEE/ACM Transactions on Networking,* 8 (5), 556–567.

[OSPF-OMP] Villamizar, C. *OSPF Optimized Multipath (OSPF-OMP).* (draft-ietf-ospf-omp-00.txt), March 1998.

[Pioro&al] Pioro, M et al. 2000. Solving an OSPF Routing Problem with Simulated Allocation. In: *Proceedings of the First Polish German Teletraffic Symposium, PGTS 2000.* Ude Verlag, 177–184.

[Smit] Shen, N, Smit, H. *Calculating IGP routes over Traffic Engineering tunnels*. (draft-hsmit-mpls-igp-spf-00.txt), June 1999.

[Thorup&al] Fortz, B, Thorup, M. 2000. Internet Traffic Engineering by Optimizing OSPF Weights. In: *Proceedings of INFOCOMM 2000.* Piscataway, NJ, IEEE, 519–528.

[Villamizar_1] Villamizar, C (UUNET). *MPLS Optimized Multipath*. draft-villamizar-mpls-omp-01.txt, February 1999.

[Villamizar_2] Villamizar, C. (November 12, 2001) [online] – URL: http://www.fictitious.org/omp/simulations.html.

IP Multiplexing for Low Capacity Links?

OLAV ØSTERBØ

In this paper we address the well known multiplexing problem in IP-networks containing links with low capacity. Due to the highly variable packet lengths real time traffic may experience transmission with unacceptable delays and jitter that may cause degraded quality.

Even if priority mechanisms are implemented in the routers (e.g. DiffServ is implemented), this will not completely solve the problem unless some kind of fragmentation of long IP packets is performed.

To study this negative multiplexing effect we have taken a non-preemptive priority queuing model, which will give the best performance for the high priority traffic classes if no fragmentation is performed. As a second model describing the effect of fragmentation we have taken a non-preemptive priority queuing model with batch arrivals where the size of a batch corresponds to the number of fragments an IP packet will consist of. The numerical examples presented show that the critical link capacity lies around 2 Mbit/s if the maximum packet length is limited to 1500 bytes.

Olav Østerbø (48) received his MSc in Applied Mathematics from the University of Bergen 1980 and joined Telenor R&D the same year. His main interests include teletraffic modelling and performance analysis of various aspects of telecom networks. Activities in recent years have been related to dimensioning and performance analysis of ATM networks and now moving towards IP networks, where the main focus is on modelling and control of different parts of next generation IP-based networks.

olav-norvald.osterbo @telenor.com

1 Introduction

Introducing high capacity links in IP-based networks, with differentiated QoS, allows delivery of services with highly variable characteristics. As we know the IP protocol provides statistical multiplexing between user applications that may generate packets of highly variable lengths. For typical real time services the main QoS parameters are end-to-end delay and jitter, and we must be aware that the main contributions to these parameters will come from low capacity links (in the access network). This justifies the need to put special emphasis on the link layer protocol and the multiplexing structure in the access network.

The access network will encompass a variety of different access technologies that are currently available. These can be divided according to

- Fixed access, or
- Mobile access.

With the recent advances in access technology the fixed access may be a mixture of one or more different types such as Asymmetric Digital Subscriber Line (ADSL), Very high speed Digital Subscriber Line (VDSL), Coax and optical fibre, all having very different physical characteristics. The logical structure of the access network may therefore be very different.

For mobile access the radio medium has limited bandwidth implying that the available bitrate for each user will be limited.

The link layer protocol structure in the access network will therefore be very different depending on the actual physical technologies applied. In Universal Mobile Telecommunication System Terrestrial Radio Access Network (UTRAN) the current link protocols are based on ATM (AAL2 or AAL5), however, there is a common trend to try to minimise the use of circuit-like protocols and instead deploy IP also in the radio network. The multiplexing of IP packets over ATM has some desired features due to the fact that the ATM cells are rather short and have fixed lengths, thereby avoiding large delay variation due to long packets.

Traditionally there has been quite a strict distinction between the access network and the core (transit) network, where the access is defined as the part of the network from the subscriber to the local exchange. By increasing the line speed by introducing different active components this definition of where the access network ends and where the core (transit) network starts are not directly valuable any more. In IP networks the definition seems to be more flexible on the basis of more functional distinctions. Usually one will define the core network as the part of the network where DiffServ and/or MPLS is deployed. By the increased line speeds it is however an interesting question to find out how ‘far’ out in the ‘old access network’ the DiffServ model (and possible MPLS) is effective.

2 A Model Evaluating the IP Multiplexing Problem for Low Capacity Links

Multiplexing traffic of different types on the IP level may cause delay and jitter problems if these traffic types share a link with rather low capacity. The main cause for this delay and jitter is the variation in the packet lengths for the different traffic types. While typical real time traffic like voice will emit packets of a small fixed size, the typical data application may generate packets that are quite long. Due to this mismatch in packet size between different applications the queuing delay for typical real time traffic may increase over the critical limit resulting in a degradation of the quality. This negative multiplexing effect will add on for each router along the path from the sender to the receiver. However, for high capacity links this queuing delay will be more or less negligible, leaving the main delay contribution to low capacity links in the access network.

One could hope that deploying the DiffServ model with traffic classification and PHB priority scheduling would overcome this problem. This is however not the case unless there is some kind of fragmentation of the long IP packets on lower layers. This means that although most of the DiffServ implementations (in routers) have implemented priority among different traffic classes these priority mechanisms are all non-preemptive. With this type of priority mechanism a high priority packet cannot interrupt an ongoing transmission of a packet of lower priority. This means that the packet length distribution of the lower priority traffic classes will have an impact on the delay for the high priority traffic.

The only way to get round the multiplexing problem for low capacity links is to have some kind of fragmentation of the long IP packet, making it possible to interleave small real time IP packets. By this option the maximum waiting time due to lower priority traffic will just be the transmission time for a single fragment. This fragmentation will be possible if IP is transported over ATM, and in this case the maximum disturbance of the high priority traffic due to lower priority is limited to one ATM cell.

In the following we shall apply two queuing models to get some quantitative experience with the problems mentioned above. The first model, without any kind of fragmentation of the IP packets, is the classical M/G/1 non-preemptive priority queuing model. The second queuing model, which includes the possibility to fragment the long IP packets is a non-preemptive priority queuing model with batch arrivals. In this model we segment the long IP packets into a batch of shorter pieces, i.e. ATM cells. As a reference model we choose the ordinary M/G/1 model. The derivation of the different performance measures such as the waiting time distribution etc. may be found in the literature and we refer to the book of Takagi [Takagi 1991] for a thorough treatment of the topic of priority queuing models.

Below we consider a link with output buffer in an IP network deploying DiffServ where we are particularly interested in the delay of the EF traffic class (high priority traffic). For simplicity we consider a model with only two priority classes:

- The EF class which is taken to be the high priority traffic;

- All other traffic which we assumed to have lower (second) priority.

Further we make the following assumptions:

- Packets arrive according to Poisson processes.

- The link capacity is C (given in bits/sec).

- The packet lengths for the high priority class is either constant or exponentially distributed with mean PL_1 (given in bits).

- The packet lengths for the low priority class are exponentially distributed with mean PL_2 (given in bits) and the length of a fragment is (constant) equal to FRL (given in bits).

- The load from the different traffic classes are ρ_1 and ρ_2 (where we assume $\rho = \rho_1 + \rho_2 < 1$).

With these definitions we get mean service times for packets and the service time for a fragment as:

$$b_1 = \mu_1^{-1} = \frac{PL_1}{C},\ \ b_2 = \mu_2^{-1} = \frac{PL_2}{C}\text{ , and}$$

$$b_f = \frac{FRL}{C}$$

In the case where the high priority traffic is exponentially distributed the Complementary Distribution Functions (CDF) of the waiting time for the highest priority traffic without any fragmentation $W_1^c(t) = P\left(W_1 > t\right)$ and with fragmentation $W_{f,1}^c(t) = P\left(W_{f,1} > t\right)$ (of the lower priority traffic) may be found to be:

$$W_1^c(t) = \frac{\rho_1}{1-\rho_1}\left(1-\rho+\frac{\rho_2}{1-\frac{\mu_1}{\mu_2}\left(1-\rho_1\right)}\right)e^{-\mu_1\left(1-\rho_1\right)t} + \frac{\rho_2}{1-\rho_1}\left(1-\frac{\rho_1}{1-\frac{\mu_1}{\mu_2}\left(1-\rho_1\right)}\right)e^{-\mu_2 t}$$

where we assume that $\mu_2 \neq \mu_1(1-\rho_1)$; and

$$W^c_{f,1}(t) = \frac{\rho_1}{1-\rho_1}\left(1-\rho-\frac{\rho_2}{(1-\rho_1)b_f\mu_1}\right)e^{-\mu_1(1-\rho_1)t}$$
$$+\frac{\rho_2}{1-\rho_1}\left(\frac{\rho_1 e^{-\mu_1(1-\rho_1)H(t-b_f)(t-b_f)}}{(1-\rho_1)\mu_1 b_f}\right.$$
$$\left.+H(b_f-t)\left(1-\frac{t}{b_f}\right)\right)$$

where $H(x) = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x < 0 \end{cases}$ is the unit step function.

The corresponding result when the highest priority traffic has constant packet lengths is more complex and includes the waiting time distribution for the M/D/1 queue as well as an integral over this distribution. (The origin of the formulas is described in more depth in Section 4.) We find:

$$W^c_1(t) = 1 - \frac{1-\rho_1-\rho_2}{1-\rho_1}q(t/b_1;\rho_1)$$
$$-\frac{\rho_2}{1-\rho_1}F(t/b_1;b_2/b_1,\rho_1)$$

where

$$q(x;\rho) = (1-\rho)\sum_{k=0}^{\lfloor x\rfloor}\frac{[\rho(k-x)]^k}{k!}e^{-\rho(k-x)}$$

is the waiting time distribution for the M/D/1 queue with service time set to unity and

$$F(t;\mu,\varphi) = \mu\int_{x=0}^{t} e^{-\mu(t-x)}q(x;\rho)dx$$
$$= \frac{\mu}{\mu+\rho}\sum_{k=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\rho+\mu}\right)^k$$
$$\left(q(t-k;\rho)-(1-\rho)e^{\mu(k-t)}\right)$$

is the convolution of the waiting time with an exponential distribution.

The corresponding result when fragmentation is performed is found to be:

$$W^c_{1,f}(t) = 1-\frac{1-\rho_1-\rho_2}{1-\rho_1}q(t/b_1;\rho_1)$$
$$-\frac{\rho_2}{1-\rho_1}\frac{b_1}{b_f}\left(G(t/b_1;\rho_1)\right.$$
$$\left.-H(t-b_f)G\left(\frac{t-b_f}{b_1};\rho_1\right)\right)$$

where

$$G(t;\rho) = \int_{x=0}^{t} q(x;\rho)dx = \lim_{\mu\to 0}\frac{1}{\mu}F(t;\mu,\rho)$$
$$= \frac{1}{\rho}\sum_{k=0}^{\lfloor t\rfloor}(q(t-k;\rho)-(1-\rho))$$

is the integral over waiting time distribution and may be obtained by taking the limit

$$\lim_{\mu\to 0}\frac{1}{\mu}F(t;\mu,\rho).$$

3 DiffServ IP Multiplexing in the Access Network?

We are especially interested in investigating the waiting time distribution when the link capacity is quite low. In the examples below we have taken the link capacity to be either 0.5 or 2.0 Mbit/sec. The high priority packet length is taken to be 200 bytes and fragmentation is based on ATM cells, i.e. 53 bytes. The load from the priority traffic is assumed to be limited to the values 0.2 or 0.3 and further the load from the lower priority traffic is taken to be either 0.5 or 0.6 giving the total load in the range 0.7 to 0.9.

In each of the figures below we have plotted four curves for the cases described, where the two lower correspond to the case where fragmentation is performed. The lowest of these is for the case where the high priority traffic has constant distributed packet lengths, and the higher is for exponentially distributed high priority packet lengths. The two highest curves (which are nearly overlapping in all the examples below) correspond to the case with no fragmentation and exponentially distributed low priority traffic. (The lowest of these nearly overlapping curves corresponds to the case where the high priority traffic has constant packet lengths, and the higher corresponds to the case where the high priority traffic has exponentially distributed packet lengths.)

Figure 1 shows that the influence of the load is rather moderate as long as the system is stable. In this example the link capacity is 2 Mbit/s. With the given parameters we observe that the waiting time distribution is nearly exponential for both cases (straight lines in the log-plot). We observe for the background traffic with packet length up to 1500 bytes only one out of 100 high priority packets will experience more than 20 ms queuing delay. So for this case there seems to be no need for packet fragmentation. However, if the packet length of the background traffic increases the picture will be different and high priority traffic will quite frequently be delayed more than 20 ms. This is shown in Figure 2.

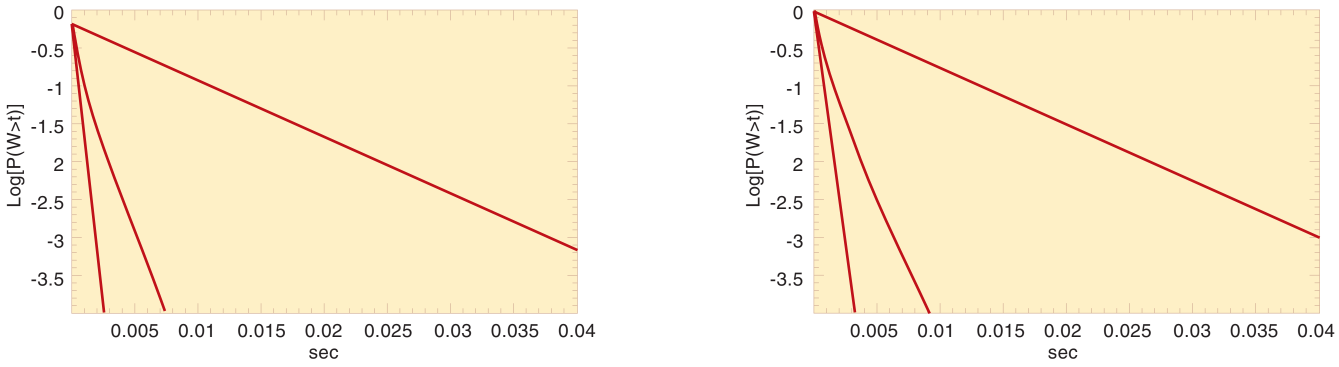


Figure 1 The complementary waiting time distribution for high priority traffic with ATM fragmentation (the lower curves) and without any fragmentation (upper curves) for a link of capacity 2 Mbit/s, and high priority packet length of 200 bytes, low priority packet length of 1500 bytes. The load is 0.2 and 0.5 in the left figure and 0.3 and 0.6 in the right figure for high priority and low priority traffic

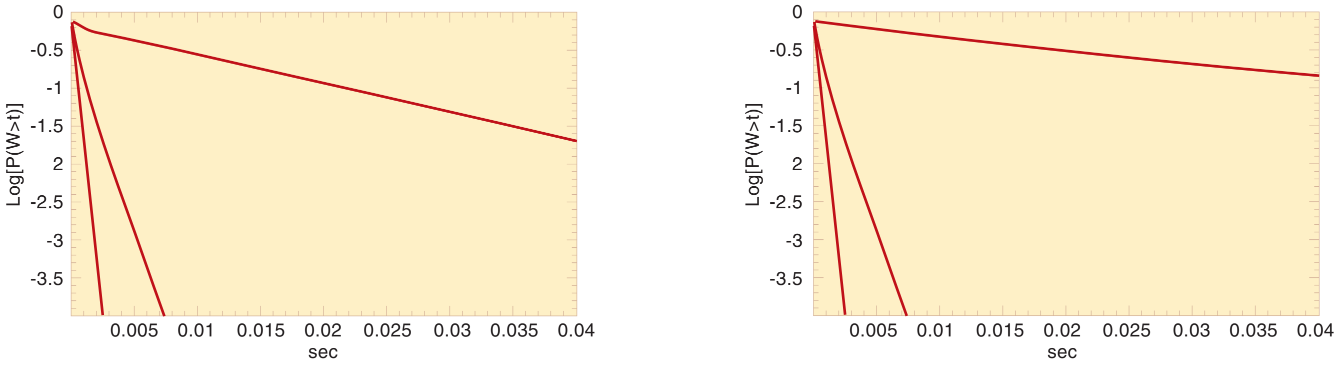


Figure 2 The complementary waiting time distribution for priority traffic with ATM fragmentation (the lower curves) and without any fragmentation (upper curves) for a link of capacity 2 Mbit/s, and high priority packet length of 200 bytes, low priority packet length of 3000 bytes in the left figure and 6000 bytes in the right figure. The load is 0.2 and 0.5 for high priority and low priority traffic

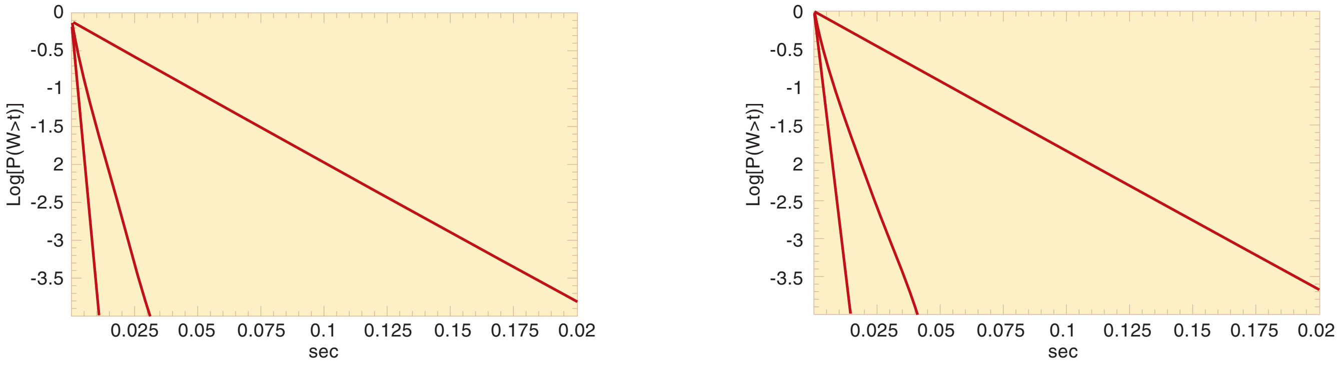


Figure 3 The complementary waiting time distribution for priority traffic with ATM fragmentation (the lower curves) and without any fragmentation (upper curves) for a link of capacity 0.5 Mbit/s, and high priority packet length of 200 bytes, low priority packet length of 1500 bytes. The load is 0.2 and 0.5 in the left figure and 0.3 and 0.6 in the right figure for high priority and low priority traffic

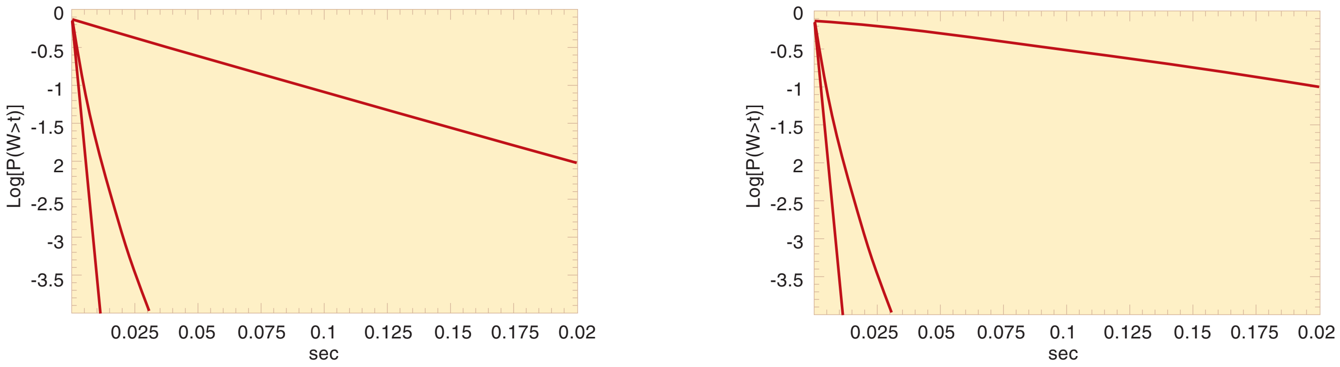


Figure 4 The complementary waiting time distribution for priority traffic with ATM fragmentation (the lower curves) and without any fragmentation (upper curves) for a link of capacity 0.5 Mbit/s, and high priority packet length of 200 bytes, low priority packet length of 3000 bytes in the left figure and 6000 bytes in the right figure. The load is 0.2 and 0.5 for high priority and low priority traffic

In the second example we have taken a slower link with capacity 0.5 Mbit/s. In this case we see that approximately one out of 100 high priority packets will get a queuing delay greater than 80 ms. This delay lies in the range where it adds up with other types of delay and may reach the limit where QoS is not possible to maintain, for instance for speech services.

If the packet length of the background traffic increases, the distribution function of the queuing delay will decrease very slowly, and with a relatively high probability the high priority traffic will experience unacceptable delays and therefore fragmentation will be necessary in order to maintain QoS.

To conclude the numerical examples it seems that the DiffServ model with the EF traffic having (non-preemptive) priority over the other classes seems to give satisfactory performance on access-links higher than 2 Mbit/s. For links with lower bitrate the multiplexing disturbance from lower priority traffic may be so high that it will be difficult to maintain stable QoS. In this case fragmentation of the lower priority traffic, for instance by deploying ATM as a link protocol, will be an efficient alternative in order to solve these multiplexing problems. This is however a critical limit since a broad part of the access links will typically be ADSL links with access rates in the range of 2 Mbit/s.

Further investigations could be done with more realistic arrival time distributions, especially with regard to the background traffic.

4 Some Methods for Calculating Delay Distributions in Non-Preemptive Priority Queues

In this section we will consider some methods to calculate waiting time distributions for priority queues. Most of the results for the M/G/1 queues with priorities are given in terms of *Laplace-Stieltjes transforms* (LSTs). To get the actual distributions we then have to invert these transforms.

We consider a *non-preemptive* queuing system with P priority classes where the priority ordering is according to the increasing numbers indexed by $p = 1, 2, ..., P$.

Packets from class p arrive according to a Poisson process with rate λ_p and the service times (denoted B_p in each class) are all independent with Distribution Functions (DF) $B_p(t) = P(B_p \leq t)$

and LST $B_p^*(s) = \int_{t=0}^{\infty} e^{-st} dB_p(t)$. We denote

mean of the service time b_p and the i^{th} moments

$b_p^{(i)}$, $i = 2, 3, ...$. Further the load from class p is given by $\rho_p = \lambda_p b_p$ and the total arrival rate and load on the server are:

$$\lambda = \sum_{p=1}^{P} \lambda_p \quad \text{and} \quad \rho = \sum_{p=1}^{P} \rho_p \ .$$

Sometimes we will also need to consider the remaining service time $\tilde{B}_p$ (from an arbitrary time until the service is finished for a given priority class). Then the Probability Density Function (PDF) of this stochastic variable is given as:

$$\tilde{b}_p(t) = \frac{1}{b_p}P(B_p > t) = \frac{1}{b_p}\int_{\tau=t}^{\infty} b_p(\tau)d\tau$$

and with LTS $\tilde{B}_p^*(s) = \dfrac{1-B_p^*(s)}{sb_p}$.

We denote W_p the waiting time for a packet of priority class p and we denote the corresponding Distribution Functions (DF) by $W_p(t) = P(W_p \leq t)$

and with LST $W_p^*(s) = \int_{t=0}^{\infty} e^{-st}dW_p(t).$

4.1 The Unsaturated Case

We consider the unsaturated case $\rho < 1$, and we define the higher priority intensity and load (from the ρ highest priority classes) by

$$\lambda_p^+ = \sum_{k=1}^{p} \lambda_k \text{ and } \rho_p^+ = \sum_{k=1}^{p} \rho_k.$$

Further we also define the service time distribution of an arbitrary packet in one of the higher priority classes denoted B_p^+ for 1, ..., p. The corresponding LST is given as the weighted sum

$$B_p^+(s) = \frac{1}{\lambda_p^+}\sum_{k=1}^{p} \lambda_k B_k^*(s)$$

with mean and ith moment given as

$$b_p^+ = \frac{1}{\lambda_p^+}\sum_{k=1}^{p}\lambda_k b_k = \frac{\rho_p^+}{\lambda_p^+} \text{ and } b_p^{+(i)} = \frac{1}{\lambda_p^+}\sum_{k=1}^{p} \lambda_k b_k^{(i)}$$

for i = 2, 3, ...

Similarly it is also efficient to define the service time distribution of an arbitrary packet in one of the lower priority classes p + 1, ..., P, which we denote B_p^-. The corresponding LST is given as the weighted sum

$$B_p^-(s) = \frac{1}{\lambda_p^-}\sum_{k=p+1}^{P} \lambda_k B_k^*(s)$$

where the rate $\lambda_p^- = \sum_{k=p+1}^{P} \lambda_k$ and corresponding load $\rho_p^- = \sum_{k=p+1}^{P} \rho_k$. Further the mean and ith moment are given as

$$b_p^- = \frac{1}{\lambda_p^-}\sum_{k=p+1}^{P} \lambda_k b_k = \frac{\rho_p^-}{\lambda_p^-} \text{ and}$$

$$b_p^{-(i)} = \frac{1}{\lambda_p^-}\sum_{k=p+1}^{P} \lambda_k b_k^{(i)} \text{ for i = 2, 3, ...}$$

We also define the remaining service time $\tilde{B}_p^-$ (of the corresponding service time B_p^-) and the LST is given by

$$\tilde{B}_p^-(s) = \frac{1}{\rho_p^-}\sum_{k=p+1}^{P} \rho_k \tilde{B}_k^*(s).$$

Based on the definitions above and by using the results found in [Takagi 1991] we may write the LST of the waiting time W_p on the following compact form:

$W_p^*(s) = W_p^+(\sigma_{p\text{-}1}(s))$ where

$$W_p^+(s) = W_{M/G/1}(s)\left(1 - \frac{\rho_p^-}{1-\rho_p^+} + \frac{\rho_p^-}{1-\rho_p^+}\tilde{B}_p^-(s)\right)$$

and where $W_{M/G/1}(s)$ is the LST of the waiting time distribution in an M/G/1 queue with input rate $\lambda_p^+ \left(= \sum_{k=1}^{p} \lambda_k\right)$ and LST of the service time given as $B_p^+(s) = \left(\frac{1}{\lambda_p^+}\sum_{k=1}^{p} \lambda_k B_k^*(s)\right)$:

$$W_{M/G/1}(s) = \frac{s\left(1-\rho_p^+\right)}{s-\lambda_p^+ + \lambda_p^+ B_p^+(s)}.$$

Further the function $\sigma_{p-1}(s)$ is defined through the LST of the busy period distribution, $\theta_{p\text{-}1}^+(s)$, generated by packets of class 1, 2, ..., p – 1:

$$\sigma_{p-1}(s) = s + \lambda_{p-1}^+ - \lambda_{p-1}^+\theta_{p-1}^+(s)$$

where $\theta_{p\text{-}1}^+(s)$ is the unique solution of the equation

$$\theta_{p-1}^+(s) = B_{p-1}^+\left(s + \lambda_{p-1}^+ - \lambda_{p-1}^+\theta_{p-1}^+(s)\right).$$

Combining the two last equations yields the important relation:

$$s = \sigma_{p-1}(s) - \lambda_{p-1}^+(1 - B_{p-1}^+(\sigma_{p-1}(s))).$$

4.2 Unsaturated Case with Batch Arrivals

To avoid the negative effect from the lower priority traffic on the high priority traffic, fragmentation of the low priority packets may be necessary in some part of the network. By cutting the long packets into a number of smaller pieces the maximum waiting time due to lower priority traffic is limited to the length of one fragment of a packet. We may model the fragmentation of packets by representing a packet arrival as an arrival of a batch of fragments (from that particular packet).

We assume that the fragments are of constant length of b_f, and the corresponding LST is

$B_f(s) = e^{-sb_f}$. With this assumption we may find the distribution of the number of fragments a packet (from priority class p) consists of as:

$g_i^p = P((i-1)b_f \le B_p < ib_f) = B_p(ib_f) - B_p((i-1)b_f)$ for $i = 1, 2, \ldots$ and further we let the corresponding generating function be:

$$G_p(z) = \sum_{i=1}^{\infty} g_i^p z^i = \sum_{i=1}^{\infty} (B_p(ib_f) - B_p((i-1)b_f))z^i;$$

and the corresponding mean value is

$$g_p = \sum_{i=0}^{\infty} B_p^c(ib_f).$$

The corresponding LST of the service time of a whole packet of class p (if it were not interrupted by fragments from higher priority packets) is then:

$$B^*_{g,p}(s) = G_p(B_f(s)) = G_p(e^{-sb_f})$$

and the corresponding mean value is $g_p b_f$. The load from the packets of class p is then $\rho_p = \lambda_p g_p b_f$. As for the case without fragmentation we define

$\lambda = \sum_{p=1}^{P} \lambda_p$ and $\rho = \sum_{p=1}^{P} \rho_p$, and we consider the unsaturated case $\rho < 1$.

We also define the higher priority intensity and load (from the p highest priority classes) by

$\lambda_p^+ = \sum_{k=1}^{p} \lambda_k$ and $\rho_p^+ = \sum_{k=1}^{p} \rho_k$. Further we also define the service time distribution of an arbitrary packet (that is not interrupted) in one of the higher priority classes $1, \ldots, p$, denoted $B^+_{g,p}$. The corresponding LST is given as the weighted sum

$$B^+_{g,p}(s) = \frac{1}{\lambda_p^+} \sum_{k=1}^{p} \lambda_k B^*_{g,k}(s)$$

with mean $b^+_{g,p} = \dfrac{b_f}{\lambda_p^+} \sum_{k=1}^{p} \lambda_k g_k = \dfrac{\rho_p^+}{\lambda_p^+}$.

Similarly we also define the rate $\lambda_p^- = \sum_{k=p+1}^{p} \lambda_k$

and corresponding load $\rho_p^- = \sum_{k=p+1}^{P} \rho_k$. We also define the remaining service time of a fragment $\tilde{B}_f$ which is uniformly distributed over the interval $(0, b_f)$ with LST:

$$\tilde{B}_f(s) = \frac{1 - e^{-sb_f}}{sb_f}$$

The queuing system described above is a non-preemptive queuing model with batch arrivals. By using the results found in [Takagi 1991] we may write the LST of the waiting time for the first fragment in a packet $W_{f,p}$ on the following compact form:

$W^*_{f,p}(s) = W^+_{f,p}(\sigma_{p-1}(s))$ where

$$W^+_{f,p}(s) = W_{M/G/1}(s)\left(1 - \frac{\rho_p^-}{1-\rho_p^+} + \frac{\rho_p^-}{1-\rho_p^+}\tilde{B}_f(s)\right)$$

and where $W_{M/G/1}(s)$ is the LST of the waiting time distribution in an M/G/1 queue with input rate $\lambda_p^+ \left(= \sum_{k=1}^{p} \lambda_k\right)$ and LST of the service time given as $B^+_{g,p}(s)\left(= \dfrac{1}{\lambda_p^+} \sum_{k=1}^{p} \lambda_k B^*_{g,k}(s)\right)$:

$$W_{M/G/1}(s) = \frac{s(1-\rho_p^+)}{s - \lambda_p^+ + \lambda_p^+ B^+_{s,p}(s)}.$$

Further the function $\sigma_{p-1}(s)$ is defined through the LST of the busy period distribution, $\theta^+_{p-1}(s)$, generated by packets of class $1, 2, \ldots, p-1$ (as for the system without batch arrivals):

$$\sigma_{p-1}(s) = s + \lambda^+_{p-1} - \lambda^+_{p-1}\,\theta^+_{p-1}(s)$$

where $\theta^+_{p-1}(s)$ is the unique solution of the equation

$\theta^+_{p\text{-}1}(s) = B^+_{g,p\text{-}1}\,(s + \lambda^+_{p\text{-}1} - \lambda^+_{p\text{-}1}\;\theta^+_{p\text{-}1}(s).$

Combining the two last equations yields the important relation:

$$s = \sigma_{p\text{-}1}(s) - \lambda^+_{p\text{-}1}(1 - B^+_{g,p\text{-}1}(\sigma_{p\text{-}1}(s))).$$

Compared with the model without fragmentation we see that the two results are similar in the way the remaining service time of the lower priority packets is introduced in the expression, however, for the model with fragmentation the influence from the lower priority traffic is limited to the remaining service time of a single fragment.

A second observation we may mention is that when the fragments are small the difference in the service times of a packet and the corresponding service times introduced by fragmentation may be small. This can be seen from the LSTs of the two variants. If we let $t_i = ib_f$ we may write

$$B^*_{g,p}(s) = G_p(e^{-sb_f}) = \sum_{i=1}^{\infty} e^{-st_i} dB_p(t_i)$$

with $dB_p(t_i) = B_p(t_i) - B_p(t_{i-1})$.

This is a well known approximation of the

integral $B^*_p(s) = \int_{t=0}^{\infty} e^{-st} dB_p(t)$ so when b_f is sufficiently small we will have $B^*_{g,p}(s) \approx B^*_p(s)$.

A major difference between the two models is that the service of a low priority packet may be interrupted after the completion of a fragment. It will therefore also be of interest to find the waiting time for the last fragment of a packet of priority class p. We denote this waiting time $W_{l,p}$. We have $W_{l,p} = W_{f,p} + D_p$ where D_p consists of the service times of all the fragments from the 'tagged' packet of class p plus the delay cycles generated from packets of the 1, 2, ..., $p - 1$ higher priority classes. The probability that a 'tagged' packet will consist of exactly i fragments

is given by $\frac{1}{g_p} i g^p_i$ so the probability that the last

fragment of a packet has exactly i fragments prior in the queue $h^{\,p}_{i}$(from that particular packet is:

$h^p_i = \frac{1}{g_p}(i+1)g^p_{i+1} \quad i = 0, 1, ...$ and the

corresponding generating function is:

$$H_p(z) = \frac{1}{g_p}\sum_{i=0}^{\infty}(i+1)g^p_{i+1}z^i = \frac{1}{g_p}G'_p(z).$$

The LST of service time distribution of all these fragments is then given as

$$B^{f^*}_p(s) = \frac{1}{g_p}G'_p(B_f(s)) = \frac{1}{g_p}G'_p(e^{-sb_f}).$$

As explained in [Takagi 1991] the corresponding LST of D_p is obtained from $B^{f*}_p(s)$ by the relation:

$$D^*_p(s) = B^{f^*}_p\;(\sigma_{p-1}(s)) = \frac{1}{g_p}G'_p\left(e^{-\sigma_{p-1}(s)b_f}\right).$$

Finally we get the LST for the waiting time of the last fragment in a packet from priority class p as:

$$W^*_{l,p}(s) = W^*_{f,p}(s)D^*_p(s)\,.$$

4.3 Waiting Time Distribution for the Highest Priority Traffic

This corresponds to the case $p = 1$ and in this case we have $\sigma_0(s) = s$, which gives the LST of the waiting time as:

$$W^*_1(s) = W_{M/G/1}(s)\left(1 - \frac{\rho^-_1}{1-\rho_1} + \frac{\rho^-_1}{1-\rho_1}\tilde{B}^-_1(s)\right)$$

where

$W_{M/G/1}(s)$ is the LST of the waiting time distribution in an M/G/1 queue with input rate λ_1 and LST of the service time given as $B_1(s)$:

$$W_{M/G/1}(s) = \frac{s(1-\rho_1)}{s - \lambda_1 + \lambda_1 B_1(s)}\,.$$

If we let $w_1(t)$ denote the density function for the waiting time we get by inverting the equation above:

$$w_1(t) = \left(1 - \frac{\rho^-_1}{1-\rho_1}\right)w_{M/G/1}(t) + \frac{\rho^-_1}{1-\rho_1}\tilde{b}^-_1(t) * w_{M/G/1}(t)$$

where $w_{M/G/1}(t)$ is the density functions for the M/G/1 queuing model and $\tilde{b}^-_1(t)$ the DF of the remaining service time for an arbitrary low priority packet.

In fact the latter formula may be explained as follows: In the long run an arriving high priority packet will find the server either idle or serving a

high priority packet with probability $1 - \frac{\rho^-_1}{1-\rho_1}$

and will 'see' the system as an M/G/1 queue, or will find the server occupied with a low priority packet with probability $\frac{\rho_1^-}{1-\rho_1}$ and will wait for the remaining service time for that low priority packet already in service plus the waiting time for an M/G/1 queue.

Of main interest is $W_1^c(t) = P(W_1 > t)$ the Complementary Distribution Function (CDF) of the waiting time. By definition $W_1^c(t) = \int_{\tau=t}^{\infty} w(\tau)d\tau$ and by integrating the relation above we get:

$$W_1^c(t) = \left(1 - \frac{\rho_1^-}{1-\rho_1}\right) W_{M/G/1}^c(t)$$
$$+\frac{\rho_1^-}{1-\rho_1}\left(1 - \tilde{b}_1^-(t) * W_{M/G/1}(t)\right)$$

where $W_{M/G/1}(t)$ is the DF and $W_{M/G/1}^{c}(t)$ is the CDF of the corresponding M/G/1 queue and $\tilde{b}_1^-(t)$ PDF for the remaining service time for an arbitrary low priority packet. More explicitly we have the following expressions for $\tilde{b}_1^-(t)$:

$\tilde{b}_1^-(t) = \frac{1}{\rho_1^-}\sum_{k=2}^{P} \lambda_k B_k^c(t),$ where

$B_k^c(t) = P(B_k > t)$ is the CDF of service time of packets from priority class k.

In the case where fragmentation is performed the results for the waiting time for the highest priority traffic looks very similar. If we let $W_{f,1}^{c}(t) = P(W_{f,1} > t)$ be the CDF of the waiting time for the first fragment of a high priority packet we find:

$$W_{f,1}^c(t) = \left(1 - \frac{\rho_1^-}{1-\rho_1}\right) W_{M/G/1}^c(t) +$$
$$\frac{\rho_1^-}{1-\rho_1}\left(1 - \frac{1}{b_f}\int_{\tau=H(t-b_f)(t-b_f)}^{t} W_{M/G/1}(\tau)d\tau\right)$$

where $W_{M/G/1}(t)$ is the DF and $W_{M/G/1}^{c}(t)$ is the CDF of the corresponding M/G/1 queue, and

$H(x) = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x < 0 \end{cases}$ is the unit step function.

4.3.1 Exponentially Distributed Service Times

To carry the results further we must assume some specific distributions. In the first (and the most straightforward) case we assume that the service times are negative exponentially distributed with mean service times μ_p^{-1} for priority class p. With these assumptions we get:

$$W_1^c(t) =$$
$$\frac{\rho_1}{1-\rho_1}\left(1 - \rho + \sum_{k=2}^{P} \frac{\rho_k}{1 - \frac{\mu_1}{\mu_k}(1-\rho_1)}\right) e^{-\mu_1(1-\rho_1)t}$$
$$+\frac{1}{1-\rho_1}\left(\sum_{k=2}^{P}\rho_k\left(1 - \frac{\rho_1}{1 - \frac{\mu_1}{\mu_k}(1-\rho_1)}\right)e^{-\mu_k t}\right)$$

where we assume that $\mu_k \neq \mu_1(1-\rho_1)$ for $k = 2, ..., P$.

The similar result when fragmentation is performed is found to be:

$$W_{f,1}^c(t) =$$
$$\frac{\rho_1}{1-\rho_1}\left(1 - \rho - \frac{\rho_1^-}{(1-\rho_1)b_f\mu_1}\right) e^{-\mu_1(1-\rho_1)t} +$$
$$\frac{\rho_1^-}{1-\rho_1}\left(\frac{\rho_1 e^{-\mu_1(1-\rho_1)H(t-b_f)}}{(1-\rho_1)\mu_1 b_f} + H(b_f - t)\left(1 - \frac{t}{b_f}\right)\right).$$

4.3.2 Constant Service Times for the Highest Priority Traffic

In this case we assume that the highest priority traffic is constant with mean $b_1 = (\mu_1^{-1})$. For the M/D/1 queue the DF of the waiting time may be written as [Roberts1996]:

$W_{M/D/1}(t) = q(t/b_1; \rho)$ with

$$q(x;\rho) = (1-\rho)\sum_{k=0}^{\lfloor x \rfloor}\frac{[\rho(k-x)]^k}{k!}e^{-\rho(k-x)}.$$

Below we show that it is possible to express the convolution with an exponential density in terms of a sum of $q(x; \rho)$'s in the following way:

$$\mu_k \int_{x=0}^{t} e^{-\mu_k(t-x)} W_{M/D/1}(x)dx = F(t/b_1; \mu_k b_1, \rho)$$

where

$$F(t;\mu,\rho) = \mu\int_{x=0}^{t} e^{-\mu(t-x)}q(x;\rho)dx =$$
$$\frac{\mu}{\mu+\rho}\sum_{k=0}^{\lfloor t \rfloor}\left(\frac{\rho}{\rho+\mu}\right)^k\left(q(t-k;\rho) - (1-\rho)e^{\mu(k-t)}\right).$$

From the last expression we also find the integral $\int_{x=0}^{t} W_{M/D/1}(x)dx = b_1 G(t/b_1;\rho)$

where

$$G(t;\rho) = \lim_{\mu\to 0} F(t;\mu,\rho)$$
$$= \frac{1}{\rho}\sum_{k=0}^{\lfloor t\rfloor}(q(t-k;\rho)-(1-\rho)).$$

If we assume that all the lower priority classes have negative exponentially distributed service times (with mean service times $b_k = \mu_p^{-1}$ for priority class p) we find by applying the formula above:

$$W_1^c(t) = 1 - \frac{1-\rho}{1-\rho_1}q(t/b_1;\rho_1)$$
$$-\frac{1}{1-\rho_1}\left(\sum_{k=2}^{P}\rho_k F(t/b_1;\mu_k b_1,\rho_1)\right)$$

If some of the lower classes have constant service times we only have to replace the term $F(t/b_1;\ \mu_k b_1,\ \rho_1)$ with the corresponding term

$$\frac{b_1}{b_k}\left(G(t/b_1;\rho_1) - H(t-b_k)G\left(\frac{t-b_k}{b_1};\rho_1\right)\right)$$

for those classes.

The result when fragmentation is performed is found to be:

$$W_{1,f}^c(t) = 1 - \frac{1-\rho}{1-\rho_1}q(t/b_1;\rho_1) -$$
$$\frac{\bar{\rho_1}}{1-\rho_1}\frac{b_1}{b_f}\left(G(t/b_1;\rho_1) - H(t-b_f)G\left(\frac{t-b_f}{b_1};\rho_1\right)\right)$$

4.3.3 Convolution of the Waiting Time in a M/D/1 Queue with an Exponentially Distributed Time

We shall use the expression

$$q(x;\rho) = (1-\rho)\sum_{k=0}^{\lfloor x\rfloor}\frac{[\rho(k-x)]^k}{k!}e^{-\rho(k-x)}$$

for the normalised waiting time for the M/D/1 queue to express the convolution

$$F(t;\mu;\rho) = \mu\int_{x=0}^{t} e^{-\mu(t-x)}q(x;\rho)dx$$

(in terms of a sum of terms $q(t-k;\rho)$ that are weighted with a geometrical factor as given above). To show this expansion we write $F(t;\ \mu;\ \rho) = \mu e^{-\mu t}\,G(t;\ \mu;\ \rho)$ where

$$G(t;\mu,\varphi) = \int_{x=0}^{t} e^{\mu x}q(x;\rho)dx =$$
$$(1-\rho)\int_{x=0}^{t}\sum_{k=0}^{\lfloor x\rfloor}e^{\mu k}\frac{[\rho(k-x)]^k}{k!}e^{-(\rho+\mu)(k-x)}dx.$$

By applying the Lemma 1 (below) we get:

$$G(t;\mu,\varphi) =$$
$$(1-\rho)\sum_{k=0}^{\lfloor t\rfloor}\int_{x=k}^{t}e^{\mu k}\frac{[\rho(k-x)]^k}{k!}e^{-(\rho+\mu)(k-x)}dx =$$
$$-\frac{1-\rho}{\mu+\rho}\sum_{k=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\mu+\rho}\right)^k e^{\mu k}\int_{\zeta=0}^{(\mu+\rho)(k-t)}\frac{\zeta^k}{k!}e^{-\zeta}d\zeta\ .$$

Integrating (and collecting) we get:

$$G(t;\mu,\rho) = -\frac{1-\rho}{\mu+\rho}\left(\sum_{k=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\mu+\rho}\right)^k e^{\mu k}\right.$$
$$\left.-\sum_{k=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\mu+\rho}\right)^k e^{\mu k}\sum_{i=0}^{k}\frac{((\rho+\mu)(k-t))^i}{i!}e^{-(\rho+\mu)(k-t)}\right)$$

The corresponding result for $F(t;\ \mu,\ \rho)$ is then:

$$F(t;\mu,\rho) = \frac{\mu(1-\rho)}{\mu+\rho}\left(\sum_{k=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\mu+\rho}\right)^k\right.$$
$$\sum_{i=0}^{k}\frac{((\rho+\mu)(k-t))^i}{i!}e^{-\rho(k-t)}$$
$$\left.-\sum_{k=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\mu+\rho}\right)^k e^{\mu(k-t)}\right).$$

Introducing the new summing variable $j = k - i$ in the first (double) sum we have $\sum_{k=0}^{\lfloor t\rfloor}\sum_{i=0}^{k} = \sum_{j=0}^{\lfloor t\rfloor}\sum_{i=0}^{\lfloor t\rfloor-j}$ and the corresponding expression may be written as:

$$(1-\rho)\sum_{k=0}^{\lfloor t\rfloor}\sum_{i=0}^{k}\left(\frac{\rho}{\mu+\rho}\right)^k\frac{((\rho+\mu)(k-t))^i}{i!}e^{-\rho(k-t)}$$
$$=(1-\rho)\sum_{j=0}^{\lfloor t\rfloor}\sum_{i=0}^{\lfloor t\rfloor-j}\left(\frac{\rho}{\mu+\rho}\right)^{i+j}$$
$$\frac{((\rho+\mu)(i-(t-j)))^i}{i!}e^{-\rho(i-(t-j))}$$
$$=(1-\rho)\sum_{j=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\mu+\rho}\right)^j\sum_{i=0}^{\lfloor t\rfloor-j}$$
$$\frac{(\rho(i-(t-j)))^i}{i!}e^{-\rho(i-(t-j))}$$
$$=\sum_{j=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\mu+\rho}\right)^j q(t-j;\rho)$$

Then collecting the results we finally get:

$$F(t;\mu,\rho)=\frac{\mu}{\mu+\rho}\sum_{k=0}^{\lfloor t\rfloor}\left(\frac{\rho}{\rho+\mu}\right)^{k}$$
$$\left(q(t-k;\rho)-(1-\rho)e^{\mu(k-t)}\right).$$

Lemma 1: Let $f_i(x)$ be a sequence of functions indexed by *i*. Then one can interchange integration and summation according to the following rule:

$$\int_{x=0}^{t}\sum_{k=0}^{\lfloor x\rfloor}f_k(x)dx=\sum_{k=0}^{\lfloor t\rfloor}\int_{x=k}^{t}f_k(x)dx\ .$$

We prove this lemma by dividing the axis into pieces between integers, and interchanging integration and summation (and collecting) as done below.

$$\int_{x=0}^{t}\sum_{k=0}^{\lfloor x\rfloor}f_k(x)dx=$$
$$\sum_{i=0}^{\lfloor t\rfloor-1}\int_{x=i}^{i+1}\sum_{k=0}^{i}f_k(x)dx+\int_{x=\lfloor t\rfloor}^{t}\sum_{k=0}^{\lfloor t\rfloor}f_k(x)dx=$$
$$\sum_{i=0}^{\lfloor t\rfloor-1}\sum_{k=0}^{i}\int_{x=i}^{i+1}f_k(x)dx+\sum_{k=0}^{\lfloor t\rfloor}\int_{x=\lfloor t\rfloor}^{t}f_k(x)dx=$$
$$\sum_{k=0}^{\lfloor t\rfloor-1}\sum_{i=k}^{\lfloor t\rfloor-1}\int_{x=i}^{i+1}f_k(x)dx+\sum_{k=0}^{\lfloor t\rfloor}\int_{x=\lfloor t\rfloor}^{t}f_k(x)dx=$$
$$\sum_{k=0}^{\lfloor t\rfloor-1}\sum_{i=k}^{\lfloor t\rfloor-1}\int_{x=i}^{i+1}f_k(x)dx+\sum_{k=0}^{\lfloor t\rfloor-1}\int_{x=\lfloor t\rfloor}^{t}f_k(x)dx+$$
$$\int_{x=\lfloor t\rfloor}^{t}f_{\lfloor t\rfloor}(x)dx=$$
$$\sum_{k=0}^{\lfloor t\rfloor-1}\int_{x=k}^{t}f_k(x)dx+\int_{x=\lfloor t\rfloor}^{t}f_{\lfloor t\rfloor}(x)dx=$$
$$\sum_{k=0}^{\lfloor t\rfloor}\int_{x=k}^{t}f_k(x)dx$$

Bibliography

Awduche, D O. 1999. MPLS and Traffic Engineering in IP Networks. *IEEE Communications Magazine,* 37 (12), 42–47.

Jamin S et al. 1997. A Measurement based Admission Control Algorithm for Integrated Services Packet Networks. *IEEE ACM Transactions on Networking,* 5 (1), 56–70.

Johnson, V. 1999. *Technology Backgrounder – Quality of Service – Glossary of Terms.* (2001, September 7) [online] – URL: www.Stardust.com.

Swallow, G. 1999. MPLS advantages for traffic engineering. *IEEE Communications Magazine,* 37 (12), 54–57.

Takagi, H. 1991. *Queuing Analysis, Volume 1: Vacation and Priority Systems, Part 1.* Amsterdam, North-Holland.

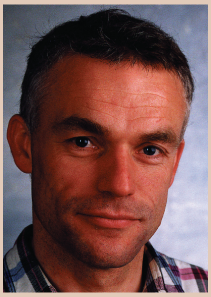
Xiao, et al. 2000. Traffic Engineering with MPLS in the Internet. *IEEE Network,* 14 (2), 28–33.

Roberts, J et al. 1996. *Broadband Network Teletraffic – Final Report of Action COST 242.* Springer.

Traffic Engineering – Inter-domain and Policy Issues

TERJE JENSEN

An essential complication when managing a network is the interconnections with other operators. This implies that the operator does not have control over the complete path, but depends on conditions in the connected networks. A further challenge is to introduce more automatic and accurate service provision, possibly adapted to individual customers and following the conditions in the network. Some means for achieving these goals are treated in this paper.



Terje Jensen (39) is Research Manager at Telenor R&D, Kjeller. He earned his PhD degree in 1995 from the Norwegian University of Science and Technology. Activities include performance modelling and analysis, dimensioning and network evolution studies.

terje.jensen1@telenor.com

1 Introduction

A network operator faced with supporting a range of services and users and being interconnected with other operators and providers, would seek ways of automating its procedures. These procedures must support efficient operation of the network while still being open for adapting services to individual users. The Traffic Engineering activities provide means for doing this. Introducing principles from the policy apparatus does further allow for effective mechanisms as more conditions can be considered during the service delivery.

The main objectives of this paper are to described challenges and some solutions for interconnecting domains and principles of policy as described by IETF. Chapter 2 illustrates the interconnection challenges in a wider scope. Some issues on the IP level and on other levels are described in Chapter 3 and Chapter 4, respectively. Chapter 5 addresses the introduction of policy, by describing the basic ideas as elaborated by IETF.

2 Interconnecting Domains

When interconnecting IP-based networks, several "levels" could be considered as illustrated in Figure 1. That is, in addition to the exchange of IP packets, interactions between management systems, service control handlers, and on the business level are expected. These have also to be considered when arranging interconnection configurations between an operator and its neighbouring actors.

On the IP level, arrangements for mapping between packet handling in the two domains have to be settled. For instance, in case of two DiffServ domains, different service classes may be defined and it must be agreed how these relate to each other. Exchange of routing information must also be agreed, like the use of routing protocols and which metrics to exchange.

The Service Level Agreements (SLAs) are attached to the business relations, although referring to Service Level Specifications (SLSs) on

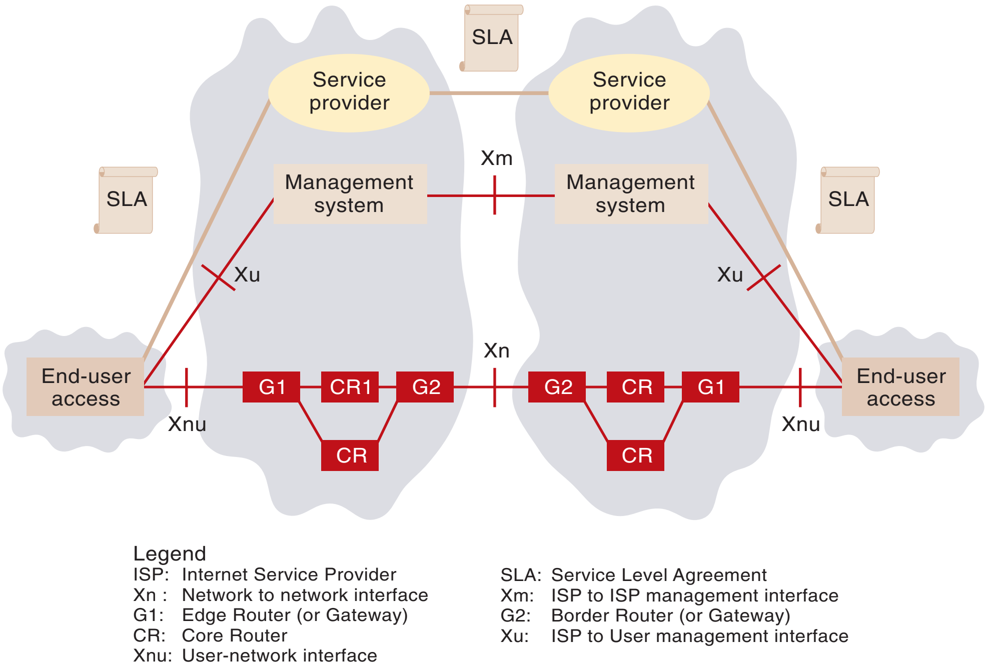

SLA
Service provider
Service provider
Xm
SLA
Management system
Management system
SLA
Xu
Xu
Xn
End-user access
G1
CR1
G2
G2
CR
G1
End-user access
Xnu
Xnu
CR
CR
Legend
ISP: Internet Service Provider
Xn : Network to network interface
G1: Edge Router (or Gateway)
CR: Core Router
Xnu: User-network interface
SLA: Service Level Agreement
Xm: ISP to ISP management interface
G2: Border Router (or Gateway)
Xu: ISP to User management interface

Figure 1 Interconnecting domains

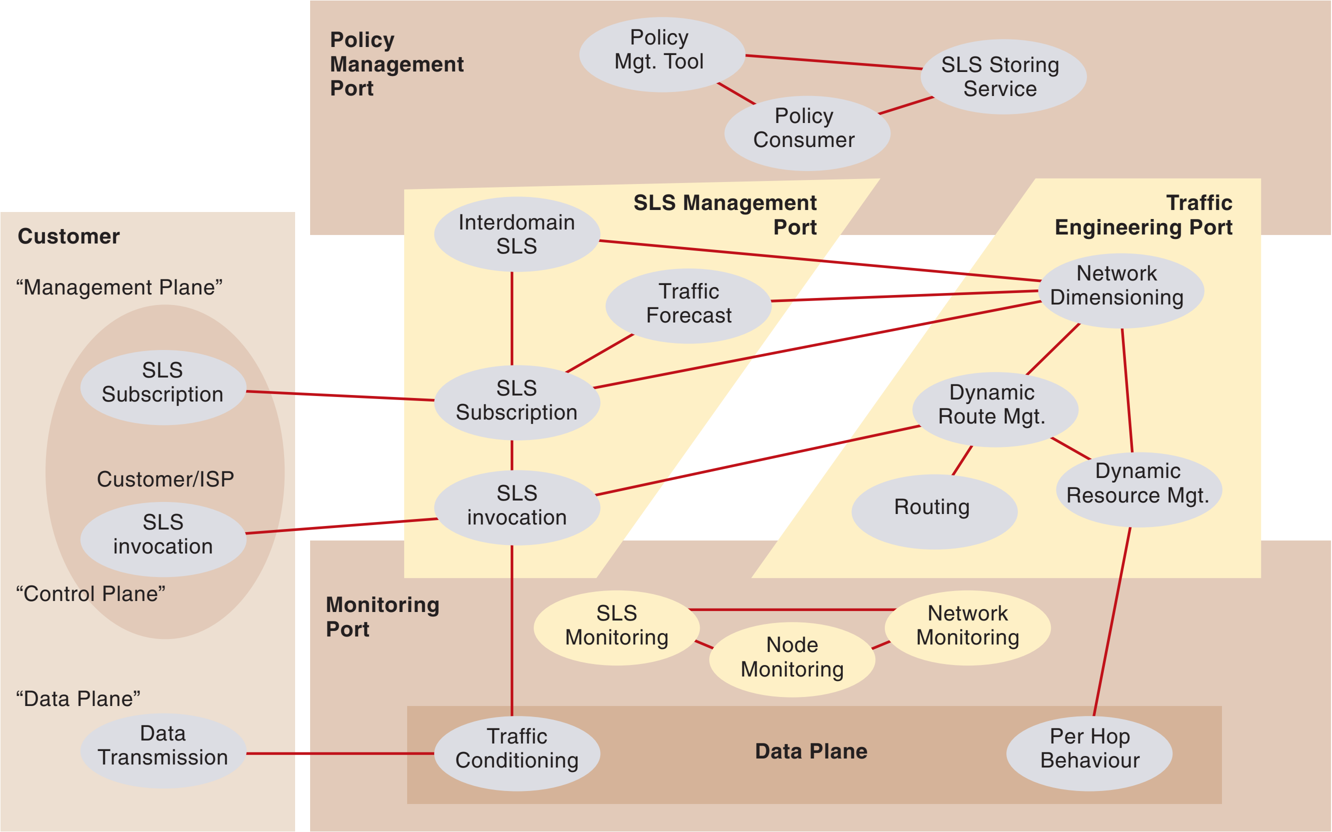


Figure 2 Potential architecture for managing IP-based services, from [Asga01]

the technical level. Means for negotiating and documenting terms in the SLAs/SLSs can be provided by the management systems. Therefore, an actor would likely eventually implement the required functionality in these systems. A potential architecture is outlined in Figure 2.

3 Issues on IP Level

3.1 Operating IntServ over DiffServ Networks

The IntServ model contains means for providing guaranteed service levels. However, the so-called scalability problem of IntServ is a main disadvantage. DiffServ has therefore been promoted, in particular in the core network where the number of flows is high. Then IntServ might be applied in the access network, in combination with DiffServ in the core portion. In order to still support end-to-end service delivery, providing IntServ across the DiffServ portion has to be promoted. A framework for this is presented in [RFC2998].

IntServ-based services are implemented by network elements, likely to be understood as routers. However, a DiffServ network "cloud" could also be seen as such a network element. IntServ contains the service classes and ways of quantifying resource requirements and deciding upon the availability of the requested resource in the network element (admission control). In order to convey this information between the network elements, RSVP has been suggested (as one candidate). In contrast to the per-flow identification used in IntServ (and RSVP), DiffServ applies a more coarse set of flows, based on the DSCP in the IP packet header. This is known as Behaviour Aggregate (BA) classification. In each DiffServ router packets are given a treatment called Per-Hop Behaviour (PHB) according to the DSCP. As DiffServ avoids per-flow processing and state information, it is said to scale better than IntServ. RSVP can then refer to aggregates.

Combining IntServ/RSVP and DiffServ can bring some benefits compared to "pure" DiffServ. One example is that admission control can be applied at the border of the DiffServ domain. Explicit signalling per flow allows for admission control, e.g. of the EF class such that the flows in that class receive the service level expected. Voice conversations are examples where admission control could be fruitful to ensure that the ongoing conversations get the service level and additional conversations are rejected in case there is not sufficient network resources.

As explicit signalling per flow is used, policy-based control, e.g. per user and per application can be introduced in a more dynamic way. Moreover, if the router in the network marks the packets, e.g. based on MF, signalling can be

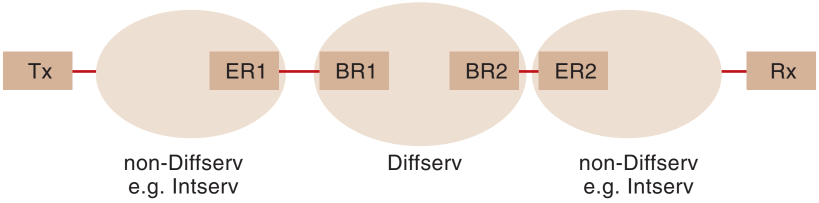


Figure 3 Reference configuration

used to convey the information to the router on which DSCP to apply for each flow. This would particularly be useful in case IPSec is applied if the IP addresses and port numbers are not statically assigned to DiffServ classes.

The reference configuration is depicted in Figure 3. The non-DiffServ regions may consist of IntServ capable routers or other types of network elements. If these regions do not handle IntServ, it is assumed that they are able to pass RSVP messages unhindered. The hosts/terminals (Tx and Rx) are able to generate and interpret RSVP-messages as they are exchanging these messages end-to-end. ER1 and ER2 are edge routers adjacent to the DiffServ region, while BR1 and BR2 are the routers connected to these within the DiffServ region.

In case the DiffServ network region is so-called *RSVP-unaware*, the edge routers (ERs) act as admission control agents to the DiffServ network. That is, they do admission control based on resource availability within the DiffServ network and on a defined policy (for instance related to Tx or Rx). For this case, routers within the DiffServ act as “pure” DiffServ routers, i.e. forward packets according to DSCP (and optionally customer policy).

If the DiffServ region is *RSVP-aware*, the border routers (BRs) apply admission control based on local resource availability and on customer (Tx, Rx) defined policy. In principle, more routers in the DiffServ region may also be RSVP aware, which means that these can also take part in the resource reservation. Still, on what granularity level, e.g. on an aggregate level, the reservation in the DiffServ region should take place, is to be decided.

At the border of the DiffServ region appropriate mapping to a PHB has to be done per flow. In addition, policing (optionally including shaping and remarking) would be needed. Admission control, taking into account the resource situation in the DiffServ region, is also necessary. The mapping could be static (from IntServ service type to DSCP) or given by information in the RSVP messages.

In order to allow for successful interconnection with a DiffServ region, that region has to meet the following requirements, ref. [RFC2998]:

- Able to provide support for the standard IntServ services between its border routers. This is to be done by invoking the PHB within the DiffServ region and appropriate behaviour at the edges of the DiffServ region. Mapping between flow characteristics in the regions must also be defined.

- Provide admission control information to the non-DiffServ network regions. This can be done by a protocol or by terms in Service Level Agreements, SLAs.

- Able to pass RSVP messages, such that it can be recovered at the egress of the DiffServ region. The DiffServ region may process these messages.

In addition, other traffic flows may be carried by the DiffServ region, e.g. not being originated in an IntServ region.

3.2 Routing Issues

Looking at most results on TE, and particularly constraint-based routing, the fact that traffic flows can traverse a number of domains is not considered. [ID_bgpte] drafts some suggestions for utilising BGP to propagate TE information between border routers. The BGP Multi-Exit Discriminator provides some level of inter-domain metric, but does not seem to include information beyond the adjacent domain. The suggestion described in [ID_bgpte] is that each domain propagates summary weights (for TE criteria).

When IGPs like OSPF and ISIS are used, the link state announcements (LSAs) can be used for carrying information from a border router to another border router informing about the metrics relevant for reaching destinations using that domain.

The TE metrics (weights) described in [IDbgpte] are:

- available bandwidth;
- unreserved bandwidth;
- colours (or class types);
- transit delay;
- IGP metrics/hops.

When a route optimisation algorithm is executed, these metrics must somehow be combined. Therefore a weight priority may also be introduced, telling which metrics are the most significant ones.

4 Interconnection – Agreements, Brokering and A^3

4.1 SLS Negotiation

Having the capability to establish SLAs *rapidly, accurately and automatically* is a significant contribution to the efficiency of a provider. This is particularly important when the number of services and customers grow.

This is argued by the ever faster evolution of the telecommunication market, leading to the introduction of more services and mechanisms. Another essential fact is that telecommunication is steadily getting more important for the customers. Hence, customers will look for service guarantees to enable them to carry out their business. Having adequate SLA-related mechanisms is therefore considered as a competitive edge by the providers/operators. As there are also dependencies between the providers, the SLAs need to be present throughout the set of providers involved, not only towards the end customer.

Handling QoS and SLA in an efficient manner introduces a number of challenges. An additional part is managing all relevant data; only a few are illustrated in Figure 4. Several non-technical aspects will also be included in an agreement between the actors. In addition to the data transfer-related aspects, issues like customer support and service provisioning will often be covered.

The SLA template is used to capture a set of Service Level Objectives for a service. A Service Level Objective is a representation of the guaranteed level of service offered. It defines an individual objective for example in terms of service metric, threshold values and tolerances. A service metric could be related to the entire service bundle, to a service element or to a single service interface, but is always related to something visible to the customer.

When the provider depends on another provider in order to fulfil the service delivery, the question arises of how to relate the interfaces and agreements as seen by that provider. This is depicted in Figure 5.

In case the individual SLAs are reflected, a scalability challenge would likely be faced.

The technical part of an SLA is called a Service Level Specification, SLS, although the actual relations between SLAs and SLSs can be more involved. A service can be said to be provided to a customer by a provider. Prior to service delivery, a negotiation would commonly take place. An example of a negotiation process is depicted in Figure 6.

The provider describes the characteristic parameters of the service to be provided, as well as any other conditions, by a Service Template Specification, STS. After deciding upon the values of the parameters, the customer returns a Service Instance Specification, SIS. This is then accepted or rejected by the provider. Any change of the service delivery conditions can trigger an update message from a provider. Then, the customer, in principle also the provider, may initiate a re-negotiation.

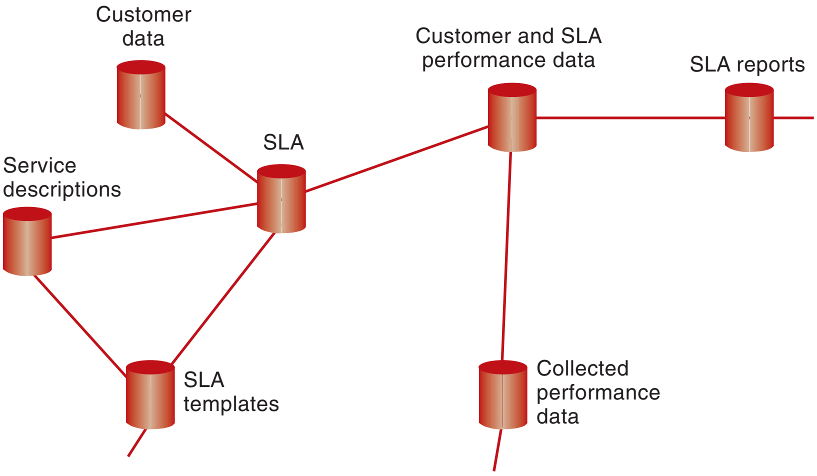


Figure 4 Some of the relevant data for managing SLAs and QoS

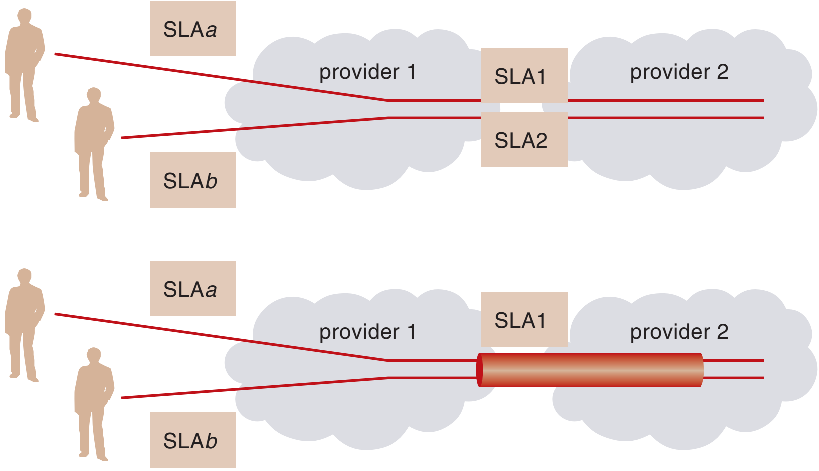


Figure 5 Reflecting individual SLAs (upper) or aggregating SLAs (lower)

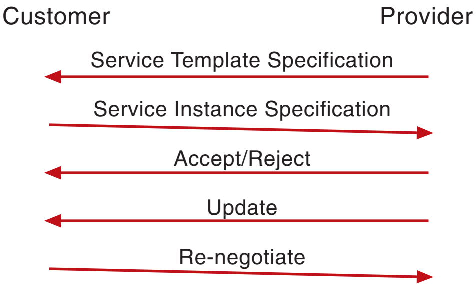

Customer
Provider
Service Template Specification
Service Instance Specification
Accept/Reject
Update
Re-negotiate

Figure 6 Example of negotiation process, [ID_sfsls]

According to [ID_sfsls], the following principles should be obeyed when designing the SLS and corresponding negotiation process:

- Different languages/protocols should be allowed.

- Negotiation at different levels and of different complexity should be supported.

- The services should not be standardised.

- The structures of STS and SIS should be simple for simple services, yet also allow for complex services.

The components of an SLS can according to [ID_sfsls] be grouped as (note that this assumes DiffServ-based services):

- Common unit: describes the terms of offering the service, e.g. identifying the provider, customer, service type, etc. The period of validity is a central component.

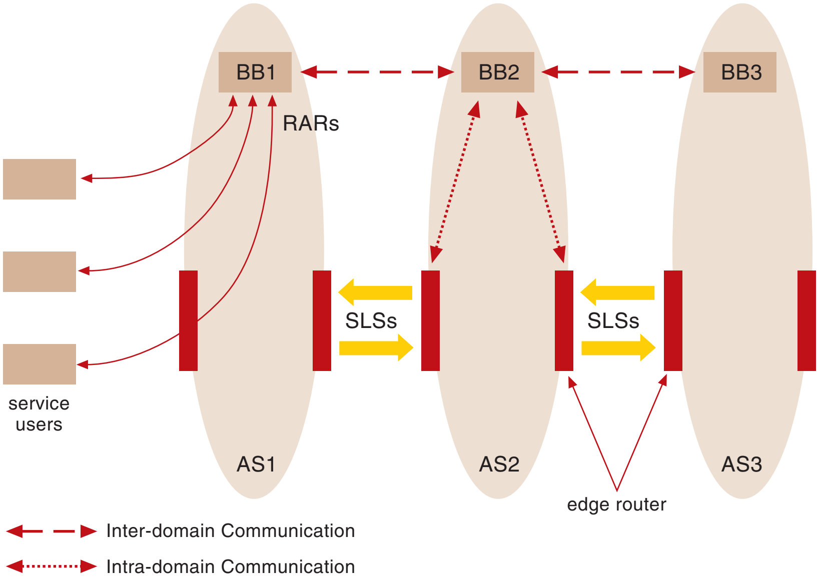
- Topology unit: describes the nature and number of end points, further divided into one Service Access Point, SAP, sub-unit and a number of graph sub-units. The SAP sub-unit gives a list of end points that specify the topology (like hose, pipe or funnel). The end points can for instance be given by IP addresses. The graph sub-unit gives a list of sources and destinations and how these are related. Unidirectional and bidirectional relations may be described.

- QoS related unit: describes the traffic flows and the service differentiation provided. Quantitative and qualitative service levels may be given for some or all parts of the topology unit. This unit may further be divided into: i) scope – giving the topology unit (graph sub-unit or end point) relevant; ii) traffic descriptor – describing the traffic flows (including DSCP, port numbers, protocol information and specification of lower layer); iii) load descriptor – gives the quantity of offered traffic, e.g. given by leaky bucket parameters, as well as treatment of excess of out-of-profile traffic; iv) QoS parameters – delay, jitter and loss for the traffic flow.

- Monitoring unit: defines a set of parameters that are to be collected and reported between the customer and provider. The structure might be similar to the QoS-related unit.

Example of a schema to apply is also included in [ID_sfsls] in addition to some selected examples.

Figure 7 Communication among BBs



4.2 Bandwidth Brokers

The Bandwidth Broker (BB) node is similar to a Policy Decision Point (PDP), see Section 5, in the sense that it makes decisions regarding bandwidth provisioning. However, bandwidth brokers tend to operate at a higher level than PDPs. PDPs are typically connected to a (small) number of Policy Execution Points (PEPs) within an administrative domain. They tend to be topology-aware as a result of their role, e.g. in the RSVP admission control process. Bandwidth Brokers are aimed more at the interfaces between domains. They tend to be less aware of the topologies within domains.

A BB refers to an abstraction that automates the admission control decisions for service requests in a network domain. This means that it is responsible for keeping track of the current allocation of reserved traffic, it is configured with policies that define which traffic flows belong to which traffic classes, and it interprets new requests in the light of these policies and the current bandwidth usage. In this sense, a BB can be considered as a special type of policy server that is responsible for those related policies for a network domain. A BB is not necessarily a policy manager but policy management and bandwidth brokering will need to work together in providing integrated policy services and admission control. Another important function of a BB is to configure network devices according to admitted QoS requests.

The concept of BB can be applied for managing intradomain and/or interdomain traffic.

In the intradomain case, the BB manages the resources based on the SLA that has been agreed upon between domains. One or more protocols are used to exchange information between a host and a BB, and a BB and a router. The BB will

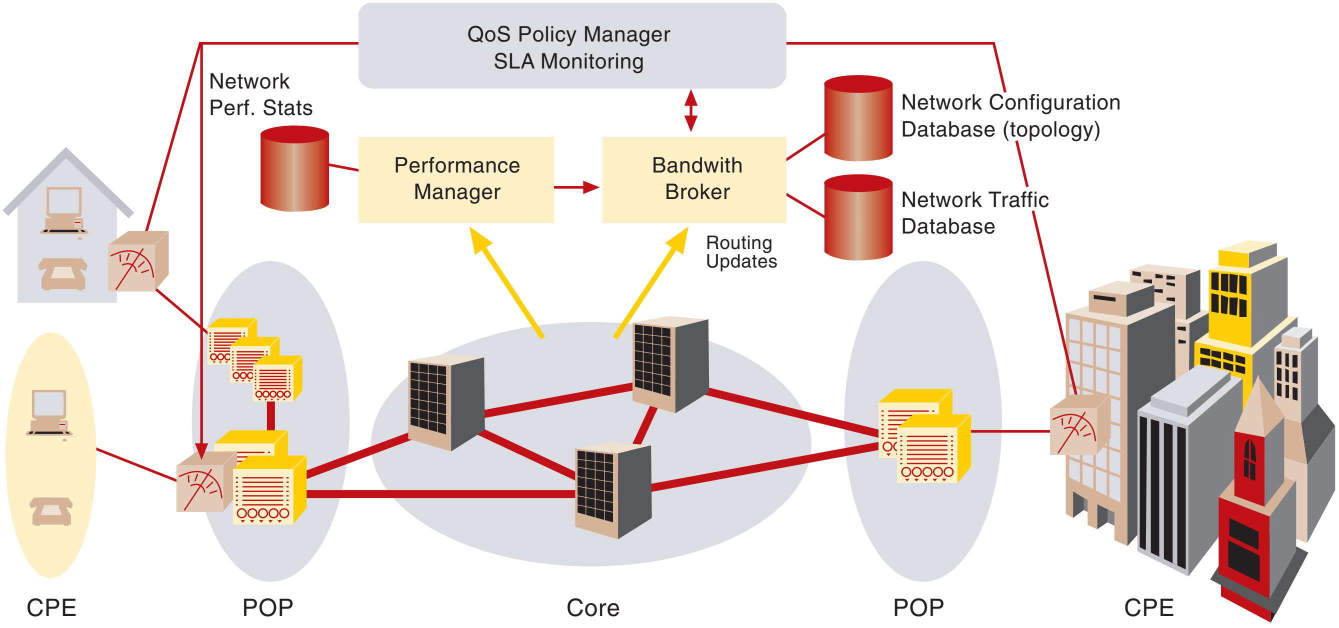


Figure 8 Location of Bandwidth Broker

communicate with the user via Resource Allocation Requests (RARs) to receive the request for bandwidth and to indicate success or failure. The BB will also communicate with the edge routers to set traffic conditioning parameters corresponding to accept reservations. Examples of protocols that can be used to communicate with routers are DIAMETER, SNMP and COPS, while protocols for communicating with hosts may include RSVP, COPS, web interfaces (HTTP) and DIAMETER.

In the interdomain case, the BB is also responsible for managing interdomain communication with BBs in neighbouring networks, for the purpose of co-ordinating SLAs across boundaries. In order to co-ordinate bandwidth assignments across domains, a single inter-domain BB protocol must exist.

Figure 7 shows a sample network configuration. It consists of three domains AS1, AS2, and AS3 with a BB for each one (BB1, BB2 and BB3). The SLAs are placed between AS1 and AS2, and between AS2 and AS3. A BB communicates with users (terminals or servers) requesting service via RARs, other BBs, and network devices (i.e. routers). In this case, a user can be either an end system or an application that requests bandwidth.

The Bandwidth Broker makes decisions based on the network topology and the network traffic characteristics. The network topology consists of a description of all available network resources: nodes, links, link metrics, physical link capacities, allocatable link capacity, resource class (gold links, links only to be used for premium customers, ...), etc. The network traffic characteristics are expressed as a set of traffic trunks, which mainly express a bandwidth requirement between core edge nodes. This information is supplied by the policy manager, which is a storage of committed SLSs.

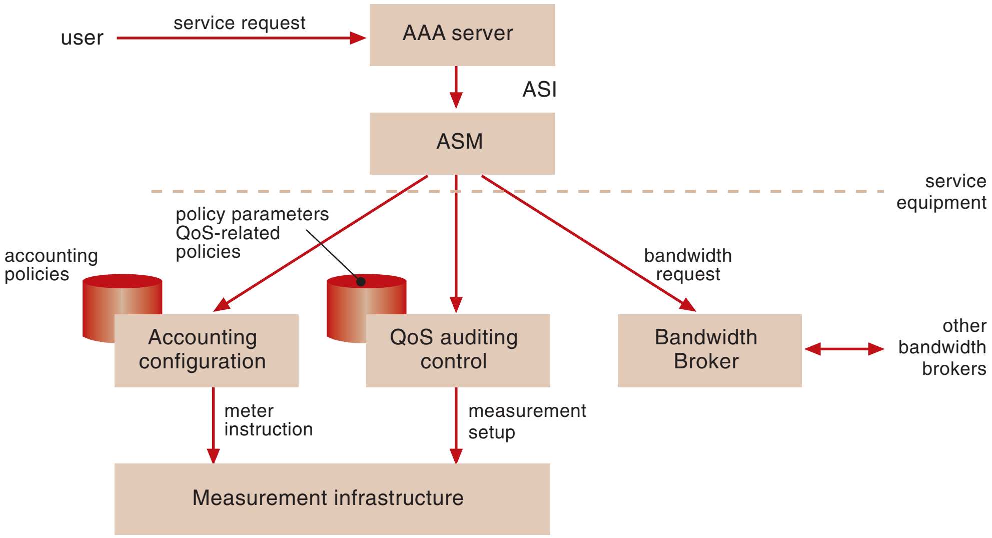
On the basis of the topology and network traffic characteristics, different indicators can be calculated by the Bandwidth Broker such as the link loads. This information can be used to determine whether a new SLS can be accepted or not. The position of the Bandwidth Broker is depicted in Figure 8.

4.3 AAA Functionality

Providing a service commercially it has to be supported by corresponding AAA functionality. The Internet Research Task Force (IRTF) has an Authentication Authorisation Accounting ARCHitecture Research Group (AAAARCH) that develops an AAA architecture. This can be said to apply a policy-based approach.

Accounting can be seen as included in the service provisioning process (called integrated accounting) or it can be offered as a separate service (called discrete accounting). In the former the accounting is coupled to a specific service, collecting relevant information by using service specific entities. A configuration for this is depicted in Figure 9. Then a common Application Specific Module (ASM) contains functions for providing the AAA services. This means that it transforms instructions from the AAA server into appropriate commands for the equipment. Relevant data on the resource usage is returned by the meters to the ASM, which compiles (conversion, aggregation, filtering) the metered data into accounting records and forwards them to the AAA server.

Figure 9 Policy-based integrated accounting, using DiffServ



To get access to a service, the user sends a service request to the AAA server. This checks the authorisation of the user and, assuming access is granted, forwards the necessary information (Application Specific Information, ASI) to the ASM. The ASM finds the information relevant for configuration of the network resources (service equipment) and distributes this information to the network nodes. In case of DiffServ, the accounting system, QoS control, and Bandwidth Broker are noted.

5 Policy and Traffic Engineering

Considering the heterogeneous network elements and traffic flows that are expected to be observed, a number of high-level requirements are placed on the management solutions:

- Automation of management task;
- Centralised management with fewer classes of management interfaces;
- Abstracted (or simplified) management data;
- End-to-end provisioning of the network;
- Consistent and uniform provisioning across all network elements;
- Standards-based solutions in order to allow inter-operability at network element and OSS level;
- Scalable solution for large networks.

The IETF Policy Management Framework has been devised keeping these requirements in mind.

5.1 Policy – What is it?

Policy can be considered as a set of principles for usage of resources, given by business considerations. That is, the business decisions are translated into statements relevant for the usage of resources in the network.

The semantics of a *policy rule* is a conditional imperative statement in the form

```
if <condition> then <action>
```

Thus, applying a rule means to evaluate its condition (matching the rule) and, depending on the outcome of that, either execute the action or not. Policy rules may be nested.

Policy-based network management would provide a centralised platform for network managers for defining and distributing network policies to enforcement points throughout a network. In a typical policy-based framework, see Figure

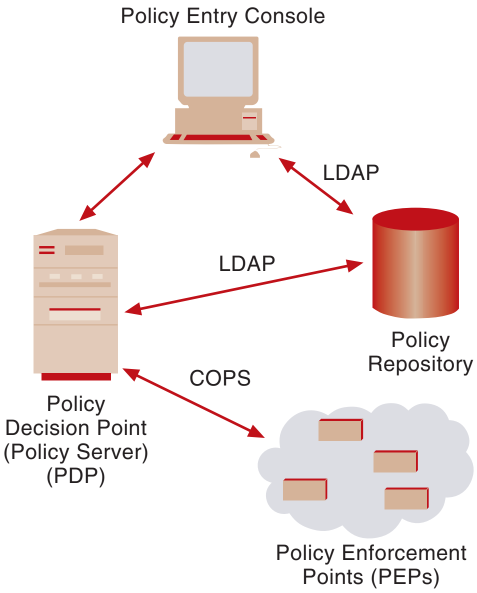

Policy Entry Console
LDAP
LDAP
Policy Repository
COPS
Policy Decision Point (Policy Server) (PDP)
Policy Enforcement Points (PEPs)

Figure 10 Policy Framework components and protocols

10, the network manager edits policies through a policy entry console. Those policies are then stored in a policy repository. When requested, a policy server (Policy Decision Point) retrieves policies from the repository and makes policy decisions that are communicated, e.g. applying Common Open Policy Service (COPS), to the relevant network points. These network points, like routers, switches and firewalls, enforce the policy decisions in the network. COPS is a query and response TCP-based protocol that can be used for exchanging information between Policy Decision Point (PDP) and Policy Enforcement Point (PEP).

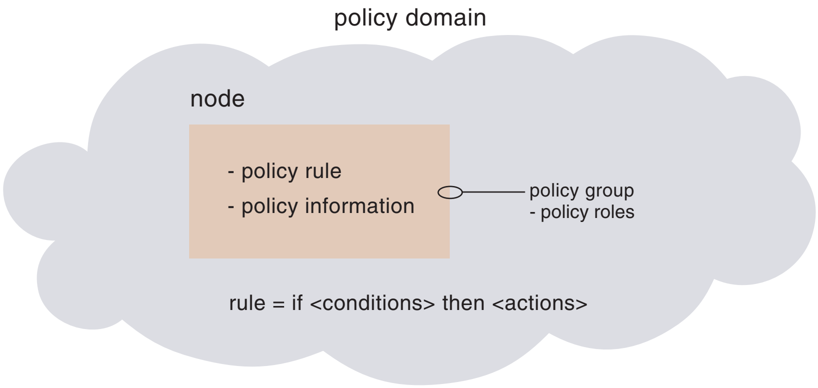


Figure 11 Policy domain and related terms

In order to carry out efficient management of the network resources a number of features of a management system are asked for, adapted from [ID_polreq]:

- Centralised management view – implying the ability to carry out management activities remotely and that the management system is able to cope with all network resources in the network.

- Abstracted management data – saying that simplified views of network resources should be possible, e.g. “hiding” details when they are not relevant.

- Common and consistent views/interfaces – similar procedures and similar data views should be used for similar procedures. The number of views/interfaces needed should also be limited.

- Automation of tasks – including less human intervention needed and that customers may serve themselves within the authorised set of activities.

A major key to meet such required features is the way of giving data used for representing the resources, customers, etc. Such data has also been referred to as *policy*. When applying policy-based management the required features listed above are addressed. So far, much effort has been spent on the representation of such data in a repository.

Having a policy repository, interfaces from the management/operator side as well as the network side have to be present. A way to transform the policy into usable formats and inform the network components also has to be implemented. Then appropriate mechanisms in the network elements to enforce the policies have to be activated.

A *QoS Policy Information Model* (QPIM) is described in [ID_qpim]. QPIM is a set of entities and relationships (both modelled by classes) that define managed objects and interactions between managed objects that can be used to define, manage and control IntServ- and DiffServ-related mechanisms using policies. Policy classes and relationships between them are depicted in Figure 12.

QPIM is an information model in the sense that it is independent of any specific implementation. The model of policies can be seen similar to an object oriented modelling in the sense that hierarchies and reuse are present. Furthermore, policies can be “nested” such that a policy contains another policy (possible decision strategies defined are match-first and match-all).

Two hierarchies of object classes are seen:

i) *structural* classes representing policy information and control of policies (entities in the managed policy environment); and

ii) *relationship* classes that show how instances of the structural classes are related to each other. Both associations and aggregations can be given as relationship classes. Containment is a directional relationship – the containing entity is known as the aggregate and the contained entities are known as the components.

A *QoS policy domain* (see Figure 11) can be viewed as a contiguous set of nodes that operate under a common system of administration and provide a common set of services. Each of the nodes can contain policy rules and/or policy information. Such a grouping is done to simplify management and ensure consistencies. A QoS policy domain can also be seen as a container that provides scooping for a QoS policy container, policy rules and other policy information.

A policy group class holds a property called policy roles. This represents the roles and combinations of roles that are associated with a set of policy rules. Each value represents one role combination. After identifying the relevant set of rules to be used, the rules have to be prioritised (e.g. first-match, match all, priority order).

The policy to apply to an entity (e.g. router) may depend on many factors such as the characteristics of the entity (e.g. protocol type used) and the user connected. To describe the functions attached to an entity, the term role is introduced, [ID_fwpib].

A role represents a functional characteristic or capability of an entity (resource) to which policies are applied. Multiple roles may be assigned to a single entity, resulting in that entity's role combination. A role should be thought of in a wider sense than an entity's attribute as the role would impact which policy is selected for an entity.

A QoS policy domain contains groups of policy rules. A policy rule can contain ordered lists of conditions and actions. The conditions and actions may be reusable objects that reside in repositories, or they may be rule-specific embedded in the rule or a combination of both. An advantage of having reusable objects is that many policy rules may refer to the same object.

One way of thinking of a policy-controlled network is to consider the network as a state machine where policies are used to control which state each of the entities should be in or is allowed to be in at any given time.

Policy rules may be aggregated into policy groups, which may be nested to represent a hierarchy of policies. Policy groups can model intricate interactions between objects that may have involved interdependencies, like application type, user identity, interface, time of day, etc. A policy group can be reused and managed as a unit. A policy rule can be called a stand-alone policy. These can be expressed as a simple statement, e.g. represented by a Management Information Base (MIB).

The set of conditions in a rule specifies when the policy rule is to be applied. The conditions can be given as sets of individual condition statements related by AND/OR. Negations can also be used. When the set of conditions associated with a policy rule is evaluated to TRUE, the set of actions in the rule is executed. This may either maintain the current state or imply a transition to another state. The order of execution of the actions can be specified.

Policy rules themselves can be prioritised, e.g. to have an overall policy with some variations in case of exceptions. An example is that *policy a)* all traffic at an interface is placed into a certain DiffServ class, except *policy b)* for packets having IP destination address equal to xxx.xxx, which are put into another DiffServ class. Then *policy b)* has to get higher priority as the actions associated with the two rules are incompatible. Hence, the exception condition gets higher priority than the general condition.

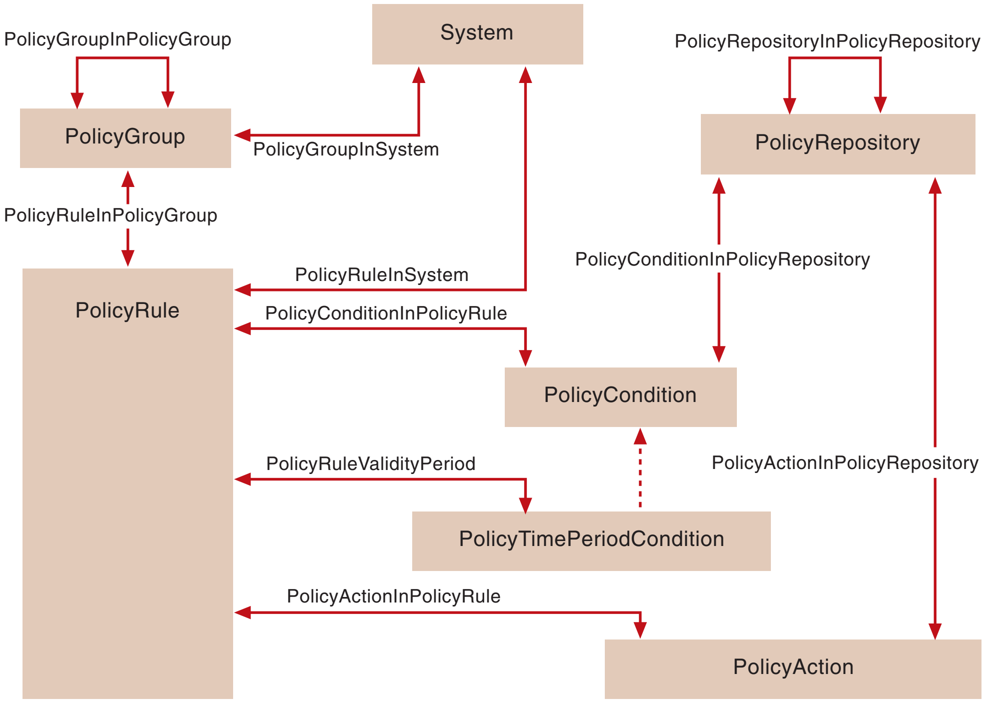


Figure 12 Policy classes and relations, from [ID_cim]

Policy rules and groups can be categorised according to purpose and intent [ID_cim], these may not be disjunctive:

- Motivational; targeting whether or how a policy's goal is accomplished, like configuration and usage policies;
- Configuration; giving the default configuration of a managed entity;
- Installation; stating what can and cannot be installed in an entity and configuration of the mechanisms that do the installation;
- Error and event; specifying which actions to undertake in case of certain events, e.g. failures;
- Usage; controlling the selection and configuration of entities based on usage data, e.g. configuration of entities for a certain traffic flow;
- Security; verifying that the user (client) is the one he claims to be and then accepting or rejecting access to entities, selecting and applying authentication mechanisms and performing accounting and auditing of entities;
- Service; characterising network and other available services.

Such a categorisation determines whether the policy is used to motivate when or how an action occurs, or to characterise services. Service policies describe services available in the network while usage policies give the binding of a user (client) to the available services.

The policies can be said to represent business goals and objectives. These goals have to be related to implementations in the network. This is described by an example of having an SLA at the higher level, which has to be related to a set of Service Level Objects (SLOs). The SLOs give the more specific metrics.

In [ID_pterm] SLA is defined as the documented results of a negotiation between a customer/consumer and a provider of a service. It specifies the levels of availability, serviceability, performance, operation or other attributes of the service. Then the Service Level Object (SLO) is defined as a partition of an SLA giving individual metrics and operational information to enforce and/or monitor the SLA. SLO may be defined as part of an SLA, or as a separate document. It is a set of parameters and their values. The actions of enforcing and reporting monitored compliance can be implemented as one or more policies.

The Service Level Specification (SLS) is related to specific handling of customers' traffic flows. It is negotiated between a customer and the provider. For DiffServ it defines a set of parameters such as DiffServ Code Points and the Per-Hop Behaviour, profile characteristics and treatment of the traffic for those Code Points. Values are also given for these parameters. An SLS is a combination of some technical part of an SLA (a negotiated agreement) and its SLOs (the individual metrics and operational data to enforce).

5.2 Policy Functions and Points

Two main elements are identified in the policy control; Policy Enforcement Point (PEP) and the Policy Decision Point (PDP), ref. [RFC2753]. These then represent the basic functions in the policy framework:

- Monitoring. The state of the network, including characteristics of traffic load and network resource, has to be estimated.
- Decision-making. This compares the current state of the network to a desired state described by an application-specific policy and decides how to achieve the desired state. PDPs are the points where policy decisions are made.
- Enforcement. This implements a desired policy state through a set of management commands; when applied to network elements, these management commands change the configuration of the device using one or more mechanisms. These mechanisms may be vendor-specific. The PEPs are the points where the policy decisions are actually enforced. It is assumed that policy decisions will always be made in the PDP and implemented in the PEP.

PEP is seen as integrated in a router, while PDP may be located in a policy server. As described in [RFC2753] a basic interaction can begin with a PEP receiving a notification/message that requires a policy decision. The PEP then formats a request and sends it to the PDP. Such a request may contain more information elements. The PDP returns the policy decision, possibly with several information elements. The PEP then enforces the policy decision, e.g. by appropriately accepting or rejecting a request and setting values for the involved mechanisms. In order to settle the policy decision, the PDP may engage other servers (e.g. using protocols like SNMP and LDAP). The PEP and PDP may be located in the same node. Furthermore, there may be policies locally stored in the node that also have to be checked (LPDP). An example of this is when an access list is stored in a border router. Then this list has to be checked in addition to

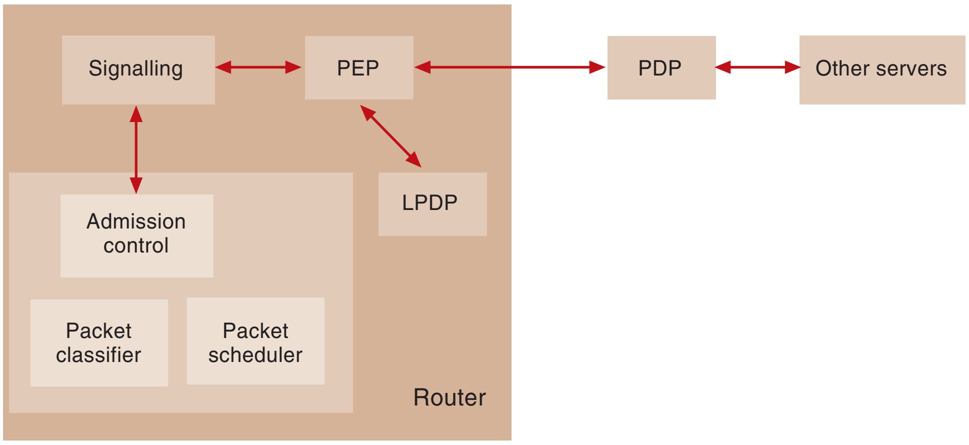


Figure 13 Relationships between policy points and other traffic control components

possibly sending a request to a PDP. This is depicted in Figure 13.

Figure 13 shows an example of relating traffic control/signalling and PEP/PDP. When a signalling message arrives at a router, the signalling module has to direct the request to the PEP. The PEP asks PDP and LPDP (PDP may override a policy given by LPDP) for decisions and returns the reply to the signalling module. Note that a PDP may also send notifications to a PEP based on other triggers, for instance to change previous decisions. In addition to PEP and PDP, reposition and management are needed, ref. Figure 14.

The Resource Allocation (RAP) Working Group (WG) is establishing a scalable policy control model for RSVP and IntServ by specifying a protocol for use among RSVP-capable network nodes and policy servers. In addition, this WG is planning to define directives for use of the Common Open Policy Service (COPS) base protocol to support policy information exchange within the framework being standardised in the IETF Policy Working Group. COPS is a query response protocol used to exchange policy information between a policy server and a set of clients.

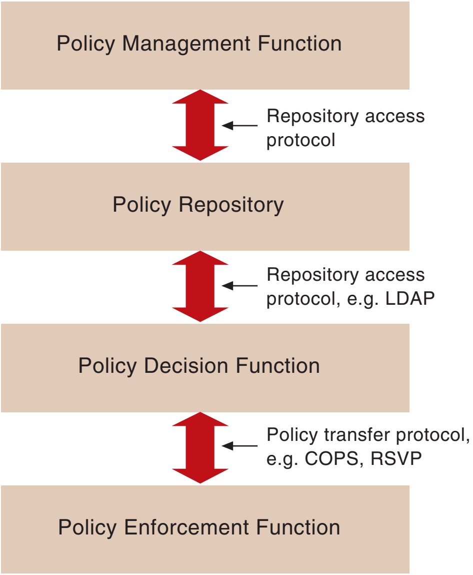

Policy Management Function
Repository access protocol
Policy Repository
Repository access protocol, e.g. LDAP
Policy Decision Function
Policy transfer protocol, e.g. COPS, RSVP
Policy Enforcement Function

Figure 14 Reference Model of Policy Framework

The PEP reports all its role combinations to the PDP in the initial COPS request message. This is also done in subsequent request messages generated in response to COPS state synchronisation requests and local configuration changes. A policy can then be given for each role combination.

COPS may also be used between Bandwidth Brokers, which essentially act as PDPs for dynamic interdomain policy exchange (see Section 5.6.1).

Policy Management Function provides the interface to the network manager. It comprises functions of policy editing, rules translation and validation. With the Policy Editor the administrator can enter, view and edit policy rules in the Policy Repository.

Once a policy rule has been entered into the Editor and before it is stored in the repository, simple validation is performed that checks for potential policy conflicts with other rules. Rule translation will resolve high level description into the specific parameters. An example is translation from names to IP addresses.

Policy Repository is a rule storage that is used for policy retrieval performed by the Policy Decision Points. The repository is also accessed in the rule validation process to detect conflicts. Access to the database is accomplished by a repository access protocol.

An architecture for QoS provisioning is described in [eTOM]. This is depicted in Figure 15.

The network management system keeps the view of the total network, including functions like:

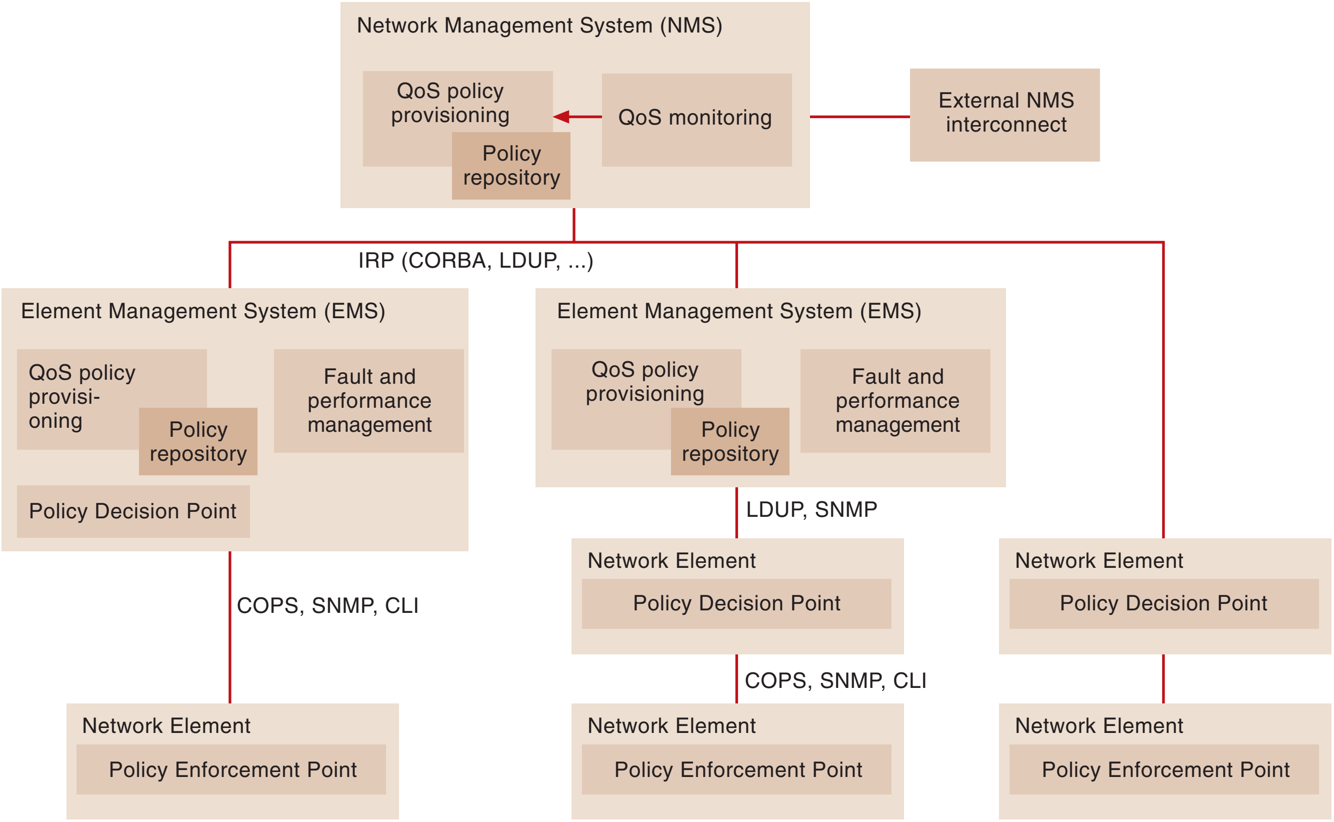


Figure 15 Conceptual architecture of policy with examples of protocols, adapted from [eTOM]

- Network policy administration user interface;

- Master network policy repository for storage of all network policies for all domains;

- Policy distribution capability to distribute policy data to the element management system policy servers;

- Global policy conflict detection.

The policy repositories may use an LDAP-based directory for storing the policy information.

The element management system QoS policy provisioning contains functions that administrate policy for a network domain. Here, a domain is an area of the network that contains equipment that performs a logically related function (e.g. access network, core network, transport network). The following functions are typically included:

- Element management system specific policy repository;

- Policy distribution capability used for distributing policy data to the PDPs;

- Local policy conflict detection.

A user interface and a PDP may also be included.

A PDP works as a policy server containing a policy repository as well as a translator converting policies from a QoS policy schema to a Policy Information Base (PIB) format. The following functions may be included:

- Domain-specific policy repository;

- Policy distribution capability to distribute policy data to PEPs;

- Translation from QoS policy schema to PIB;

- Optional real-time policy decision-making function;

- Local policy conflict detection.

The PEPs implement the policies, incorporating functions like:

- Storage of policy-related data in its MIB;

- Execution of policies according to state and events.

The QoS monitoring contains functions for collecting, processing performance statistics, usage data and QoS related faults. This includes functions like:

- Manage QoS fault conditions received from network elements;

- Retrieve QoS performance data from network elements;

- Collect and process usage data;

- General QoS reports – trend analysis of key QoS parameters;

- Audit/analyse collected QoS parameters against expected values.

These are typically distributed on functionality in the network elements, element management system and network management system.

5.3 Policy Actions

Three types of actions are defined in order to control QoS enforcement:

- Signalling used to interact with RSVP. Signalling-related policies are related to admission control (e.g. whether to accept or reject a request arriving by RSVP), controlling the forwarding behaviour (e.g. set appropriate marking) and the signalling procedures (e.g. setting and modifying values in the RSVP messages).

  In order to utilise RSVP, a few additional updates of objects have been described, ref. [RFC2750]. They are called policy data objects. The Filter_spec and Scope objects describe the associated senders and prevent loops. RSVP_hop identifies the neighbour policy-capable router, both an originating and a destination may be given. The Integrity object may provide a secure communication channel between non-adjacent PEPs, i.e. without involving Policy-Ignorant Nodes (PINs). The Policy_refresh object can be used to set values for when the policy association has to be refreshed, e.g. for authentication.

- Provisioning used to enforce differentiated service policies including marking, policing and shaping. Meters measure the temporal properties of a flow (or flow aggregate) of packets selected by a classifier against a traffic profile. Meters measure flows matching the rule condition per flow, per interface, per role within a device, per device or per role across all devices. Traffic can be classified as conforming, excess or violating. The measurement value may possibly also be compared against a profile. For instance, a shaper, policer and re-maker compare a traffic profile against a meter. Common parameters seen refer to rate (e.g. in kbit/s), normal burst (e.g. in byte) and excess burst (e.g. in byte).

  Markers are used to assign a DSCP to a packet. This could be done based on the state of the meter referring to the traffic flow that the packet belongs to. A shaper is used for delaying packets in a traffic flow to bring the flow into conformance with a profile. Droppers are used to discard some of the packets in a traffic flow, commonly to bring the flow in conformance with a profile (frequently referred to as policing). These are recognised from the functional elements described for DiffServ.

  An example of a policy is

  i) if "traffic flow within profile *X*" then "mark packet with DSCP = AF1"

  ii) if "traffic flow out of profile *X* and within profile *Y*" then "mark packet with DSCP = AF2"

  iii) if "traffic flow out of profile *Y*" then "drop packet"

  where profile *X* can mean rate is less or equal to 64 kbit/s, and profile *Y* can mean rate is less or equal to 128 kbit/s.

  This could be done more efficiently when condition ii) is only checked if i) fails; condition iii) is only checked if ii) fails.

- Per-Hop-Behaviour (PHB) used to enforce the behaviours across a DiffServ domain. PHB actions are used to represent the requirements on PHBs and also giving details enabling mapping onto configuration parameters for configuring queues, schedulers, droppers and other mechanisms, i.e. including attributes related to DiffServ MIBs. These involve setting bandwidth to be allocated, delay and jitter parameters, use of dropping algorithm with corresponding values, etc. These can be specified as hierarchical policies, i.e. when certain rules are valid for an aggregate (e.g. all packets on a given interface) while further rules are to be obeyed for traffic flows within the aggregate (e.g. for TCP packets on the same interface).

All these types may be included in a single action/rule.

The IETF Policy Working Group is standardising the basic framework of policy-based management systems for IP networks. It focuses on representing, managing and sharing policies in a vendor independent, interoperable and scalable manner.

The Policy WG co-ordinates the development of the QoS schema with the Policy Information Base (PIB) and the Management Information

Base (MIB) being developed in the DiffServ WG as well as with extensions to the COPS being developed by the Resource Allocation Protocol (RAP) WG.

Policy rules must be represented as data structures so they can be stored and retrieved. To address this issue, the IETF 's Policy Working Group has defined the Policy Framework Core Information Model, which defines a high-level set of object-oriented classes that can be used for general policy representation. The intent of the Policy Working Group of the IETF is closely related with the work underway in the Directory Enabled Network (DEN) Working Group, and in the Networks Working Group of the Distributed Management Task Force (DMTF). As a result, the DEN standards have been adopted by the DMTF as part of their Common Information Model (CIM) and CIM itself serves as the basis for the IETF Policy Working Group's core model.

5.4 Policy-enabled MPLS Networks

In general, policy management for MPLS involves Life Cycle management (i.e. creating, deleting and monitoring) of Label Switched Paths (LSPs) through the network along with controlling traffic flow admission (LSP Admission Control) to those managed resources. MPLS supports explicit traffic engineering via a number of specifications (CR-LDP, RSVP) that allow LSPs to be managed based on certain constraints. The policy management architecture used to control traffic engineering functionality should be independent of the MPLS mechanisms used. An objective of introducing policy management is to arrive at predictable network services.

A major application of MPLS is in providing traffic engineering capabilities to IP networks. In some cases, this may involve the use of specific mechanisms (e.g. DiffServ- and IntServ-related).

For handling MPLS related to traffic engineering, there are two basic categories of polices, ref. [ID_MPLScops]:

- LSP/tunnel management (or LSP life-cycle) policies; dealing with configuration related to initiating, maintaining and removing LSPs;

- LSP admission control (or flow management) policies; dealing with classification for mapping traffic flows onto LSPs.

In an MPLS environment, the PEP resides in the LSRs (Label Switching Routers). A connection to a policy server (PDP) is then required, although several of the decisions would likely be taken internally in the LSR, possibly notifying the policy server. However, allowing the PDP to make the decisions may result in a more efficient operation for the total network. Then a Request message (REQ) may be sent from the PEP to the PDP when an LSP is to be established, e.g. due to an incoming RSVP or CR-LDP message to the LSR. The PDP will reply with a Decision message (DEC) instructing the PEP on how to set up the LSP. During the operation a Report message (REP) can be used by the PEP to acknowledge the DEC and report performance-monitoring results.

5.5 Differentiated Services Policy Information Base

The DiffServ Working Group has also released an Internet Draft specifying a set of Policy Rule Classes (PRCs) designed for configuring QoS policy for Differentiated Services. The base module contains the PRCs for setting DiffServ policy queues, classifiers, meters, etc., and also contains filters for matching IP packets.

This may be broken down into several different groups, including:

- QoS Interface Group: contains PRCs which can be used to tell a PDP the types of interface supported by a PEP and the PRCs a PDP may install to configure the PEP. Examples of attributes are: queues, scheduling parameters, buffer sizes, etc.

- QoS Metering Group: contains PRCs relating to the configuration of meters.

- QoS Action Group: contains PRCs used to define the actions to be taken after the result of classification and metering. It also contains policies that associate classifiers, meters and actions.

- IP Classification and Policing Group: contains policies that define IP classifier elements.

5.6 Some Related Protocols

5.6.1 COPS

The main characteristics of the COPS protocol include:

- A client/server model where the PEP sends requests, updates, and deletes to the remote PDP and the PDP returns decisions back to the PEP.

- The utilisation of TCP as its transport protocol for reliable exchange of messages.

- The protocol is extensible in that it is designed to take advantage of self-identifying objects and can support diverse PEP specific information without requiring modifications to the COPS protocol itself.

- COPS provides message level security for authentication, replay protection, and message integrity. COPS can also reuse existing protocols for security such as IPSec to authenticate and secure the channel between the PEP and the PDP.

- The protocol is stateful in two main aspects:

  - Requests from PEP are installed or remembered by the remote PDP until they are explicitly deleted by the PEP. At the same time, decisions from the remote PDP can be generated asynchronously at any time for a currently installed request state.

  - The PDP may respond to new queries differently because of previously installed Request/Decision state(s) that are related.

- Additionally, the protocol is stateful in that it allows the PDP to push configuration information to the PEP, and then allows the PDP to remove such state from the PEP when it is no longer applicable.

Note that the COPS architecture does not provide a complete management framework as such. It merely provides a way to distribute policy configuration information to devices. The COPS architecture relies on other management protocols, e.g. for monitoring.

A client-type of COPS for TE is drafted in [ID_COPS]. There, within an IP router and in addition to a PEP and the LPDP, a set of routing information bases (RIBs) and Forwarding Information Bases (FIBs) are also defined. The RIB represents a routing protocol, like OSPF and BGP. The FIB stores the routes that have been selected by the routing processes. The request, decision and report messages have to include relevant attributes for traffic engineering, like link metrics and traffic flow characteristics.

5.6.2 LDAP

While there are many choices of protocols for directory/database access from the policy management function and policy decision function, LDAP appears to be favoured by a number of vendors and users. LDAP schemes are versatile and allow considerable flexibility in the choice of back-end directory management. Further, the LDAP client-server protocol is widely implemented and used for supporting a wide range of directory enabled applications. However, there is a number of shortcomings of LDAP that must be clearly understood by implementers, such as lack of asynchronous notification, replication support, security, referential integrity, support for "templates", and limitations of query language. Some of these shortcomings, such as asynchronous notification, may be addressed by defining specialised protocols between functional entities. LDAP is further described in [RFC2251].

5.6.3 SNMPv3

The original SNMPv1 was a lightweight management protocol that was sufficient for small networks offering a best effort service. Its capability was generally limited to the monitoring of network element operation and performance rather than performing intrusive management operations.

As data networks and their applications have grown, so has the realisation that they must be able to offer similar quality, availability and scalability guarantees as are provided for classical networks. Consequently, SNMPv3 is being developed to meet these requirements. Furthermore, SNMP usage has become more strategically important for operators as IP technology is deployed in their networks. SNMPv3 is further described in [RFC2570], [RFC2571], [RFC2572] and [RFC2574].

An IETF SNMP working group is to elaborate, among other things, some QoS MIB modules to describe management objects for the control of DiffServ policy in co-ordination with the effort currently taking place in the DiffServ WG.

In addition, the DiffServ WG is producing an MIB designed according to the DiffServ implementation conceptual model. The purpose of the DiffServ MIB is to allow the setting up of Multi-Field and Behaviour Aggregate traffic classification filters and queues; to monitor whether or not a traffic flow is within its profile; and finally to perform some action on the traffic depending on whether or not it is in profile (shaping, policing, (re-)marking). SNMP is the protocol used to implement the DS MIB. The MIB is composed of six basic elements:

- Behaviour Aggregate Classification table – stores DSCPs in order to enable identification of tagged streams of traffic. Could be part of the Classifier Table, but for extensibility reasons is kept as a separate table. Note that a new draft of the DiffServ MIB has merged this filter with the Multi-field now described.

- Multi-Field Classification table – used to define MF Classifiers. Similar to the BA Classification Table, this could be part of the Classifier Table but it has not been specified in that way. This permits other proprietary filters to be specified, bringing the need for just one classifier table, thus simplifying management of the information.

- Classifier table – indicates how traffic flows are to be sorted. The criteria used here could in theory be any identifiable property of a particular flow or behaviour aggregate (DSCP).

- Metering table – implemented as a simple set of pass/fail tests applied to a stream of traffic passing a particular token bucket meter. The action to be taken with conforming and non-conforming traffic is specified in the Action table. It is also possible to cascade the meters in this table for more complex behaviour.

- Action table – Several actions are considered in this specification: traffic marking, counting of the traffic passing a certain point, applying a drop policy, queueing of traffic. The elements in this table specify behaviour resulting from a classification, a metering operation or another action.

- Queue table – This table specifies the behaviour of individual queues, in terms of bandwidth and queuing mechanisms. The elements specified in this table are the result of a queueing action and can be used for both queuing and shaping.

6 Concluding Remarks

Efficiently managing the IP-based network is an obvious goal for an operator. Searching for this, the operator would utilise different mechanisms referring to the IP level, the management system and on the business level. This paper has outlined some of the essential aspects that should be considered, relating Traffic Engineering to management systems, policy and Service Level Agreements.

References

[Asga01] Asfari, A et al. 2001. A Monitoring and Measurement Architecture for Traffic Engineering IP Networks. In: *Proc. of IEEE/IFIP/ IEE Int. Symp. on Telecom. (IST2001).*

[eTOM] 2001. TeleManagement Forum GB921: *eTOM – The Business Process Framework. Ver. 1.0.*

[ID_bgpte] Abarbanel, B, Venkatachalam, S. 2000. *BGP-4 support for Traffic Engineering.* draft-abarbanel-idr-bgp4-te-01.txt. Work in progress.

[ID_cim] IETF. 2000. Moore, B et al. *Policy Core Information Model – Version 1 Specification.* draft-ietf-policy-core-info-model-08.txt. Work in progress.

[ID_COPS] Jacquenet, C. 2001. *A COPS client-type for IP traffic engineering.* draft-jacquenet-ip-te-cops-02.txt. Work in progress.

[ID_fwpib] IETF. 2000. Fine, M et al. *Framework Policy Information Base.* draft-ieft-rap-frameworkpib-03.txt. Work in progress.

[ID_MPLScops] Reichmeyer, F, Wright, S, Gibson, M. 2000. *COPS Usage for MPLS/Traffic Engineering.* draft-franr-mpls-cops-00.txt. Work in progress.

[ID_polreq] IETF. 2000. Mahon, H et al. *Requirements for a Policy Management System.* draft-ietf-policy-req-02.txt. Work in progress.

[ID_pterm] IETF. 2000. Westerinen, A et al. *Policy Terminology.* draft-ietf-policy-terminology-01.txt. Work in progress.

[ID_qpim] IETF. 2000. Snir, Y et al. *Policy Framework QoS Information Model.* draft-ietf-policy-qos-ifno-model-02.txt. Work in progress.

[ID_sfsls] Rajan, R et al. 2000. *Service Level Specification for Inter-domain QoS Negotiation.* Work in progress.

[RFC2251] IETF. 1997. Wahl, M, Howes, T, Kille, S. *Lightweight Directory Access Protocol (v3).* (RFC 2251.)

[RFC2570] IETF. 1999. Case, J et al. *Introduction to Version 3 of the Internet-standard Network Management Framework.* (RFC 2570.)

[RFC2571] IETF. 1999. Harrington, D et al. *An Architecture for Describing SNMP Management Frameworks.* (RFC 2571.)

[RFC2572] IETF. 1999. Case, J et al. *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP).* (RFC 2572.)

[RFC2574] IETF. 1999. Blumenthal, U, Wijnen, B. *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3).* (RFC 2574.)

[RFC2750] IETF. 2000. Herzog, S. *RSVP Extensions for Policy Control.* (RFC 2750.)

[RFC2753] IETF. 2000. Yavatkar, R et al. *A Framework or Policy-based Admission Control.* (RFC 2753.)

[RFC2998] IETF. 2000. Bernet, Y et al. *A Framework for Integrated Services Operation over Diffserv Networks.* (RFC 2998.)

Agreements in IP-based Networks

IRENA GRGIC AND METTE RØHNE

Irena Grgic (30) is Research Scientist at Telenor R&D, Kjeller. She is mainly involved in activities related to QoS and charging for different networks and systems, and studies related to network evolution, both in international and national projects. She holds an MSc in Electrical Engineering from the University of Zagreb in 1999. She was Task Leader of Task 5 in EURESCOM P806-GI.

irena.grgic@telenor.com

Mette Røhne (35) is Research Scientist at Telenor R&D, Kjeller. Her main activities include applied QoS, network design and techno-economic studies, performed both in international and national projects. She received her PhD degree in 1999 from the Norwegian University of Science and Technology.

mette.rohne@telenor.com

1 Introduction

The telecom market is nowadays characterised by steadily increasing complexity and dynamic changes. There are multiple causes, like users whose technical knowledge and demands are increasing, applications ask for high quality of services, the number of services and the number of providers offering these are getting larger, a variety of technologies are used. Also, business models are changing in that new roles are present and multiple providers are taking those roles. In order to differentiate themselves in such a market, providers are aiming at attracting the users by offering services with assured Quality of Service (QoS).

Traditionally, QoS is a very important element of the service offer for users. Assuring QoS requires a provider to study, understand and handle both business and technical aspects in a consistent way. It is not sufficient to understand both of these issues separately, but they should rather be observed and studied simultaneously. Providing services with assured QoS to the users with ever-increasing demands for services crossing multiple domains administrated by different providers sets challenges on these providers. Simply, in order to fulfil users' demands end-to-end providers have to co-operate while at the same time competing for the same market segment. Hence, the need to describe principles for arranging relationships between providers is steadily getting more pronounced. Generally speaking, any relationship between two actors is associated with a set of expectations as well as a set of obligations. These expectations and obligations may be implicit, but it is better to have them explicitly agreed, especially in a business context. Various types of agreements present in today's telecom market, and their relationships are discussed in this paper, but the most pronounced type is certainly a Service Level Agreement (SLA). Briefly, an SLA is an agreement between two parties that deals with the level of service to be delivered. The SLA has two main parts covering business and technical aspects. Technical aspects and QoS-related issues are the focus of in this paper. The technical part of an SLA includes one or more Service Level Specifications (SLS). An SLS is a specification that envelopes a set of parameters and their values that are specified for a service provided to traffic flow. Mapping between SLAs and SLSs is not plain and straightforward, as will be discussed later.

The situation where services are supported by the infrastructure based on the Internet Protocol (IP) technology is even more complex, since the technology itself, i.e. different aspects and mechanisms, are not yet mature. On the other hand, the simplicity and transparency of the IP allow for a high dynamics factor, that implies e.g. a variety of services appearing very fast, a variety of roles taken by the providers (and relationships between them) that can be changed easily, and this situation can be described as a multi-service multi-provider environment. Assuring QoS in such an environment is challenging many providers – therefore, the issues of settling SLAs is getting more pronounced. Apart from assuring QoS, handling and assuring SLAs in an IP-based multi-service multi-provider environment is not trivial. Some of the issues to help better understanding and handling of SLAs and their aspects in general and in an IP-based environment, in particular, are addressed in this paper.

Settling SLAs between all parties involved in the service provision/usage enables assurance of the QoS for user traffic crossing several domains. Also, the process of designing SLAs is not a trivial task for a provider. Taking perspective of the provider (Figure 1), numerous data are relevant as the input to the process of designing SLAs, negotiating them and finally realising them.

Negotiation of an SLA can be initiated either by the user who has their requirements, or by the provider who is offering its services. Both parties should collect relevant input information before negotiating the SLA. As illustrated in Figure 1, from the provider's perspective, the input includes the knowledge of the business model/strategic decisions, core business description and focus, service portfolio description, technical infrastructure, charging schemes, SLA/SLS monitoring, QoS parameters, and mechanisms locally implemented in the provider's domain. As the input to the negotiation of the QoS-part of the SLA, the service description and scenario have to be available, the list of desired objectives for the particular QoS characteristics (e.g. parameter values) has to be indicated by the user, and a list of potential subproviders should be obtained. By running the procedure[1)], the provider would have to make decisions on the trade-off between the degree of supporting the mechanisms locally in his domain, and the degree to which service compo-

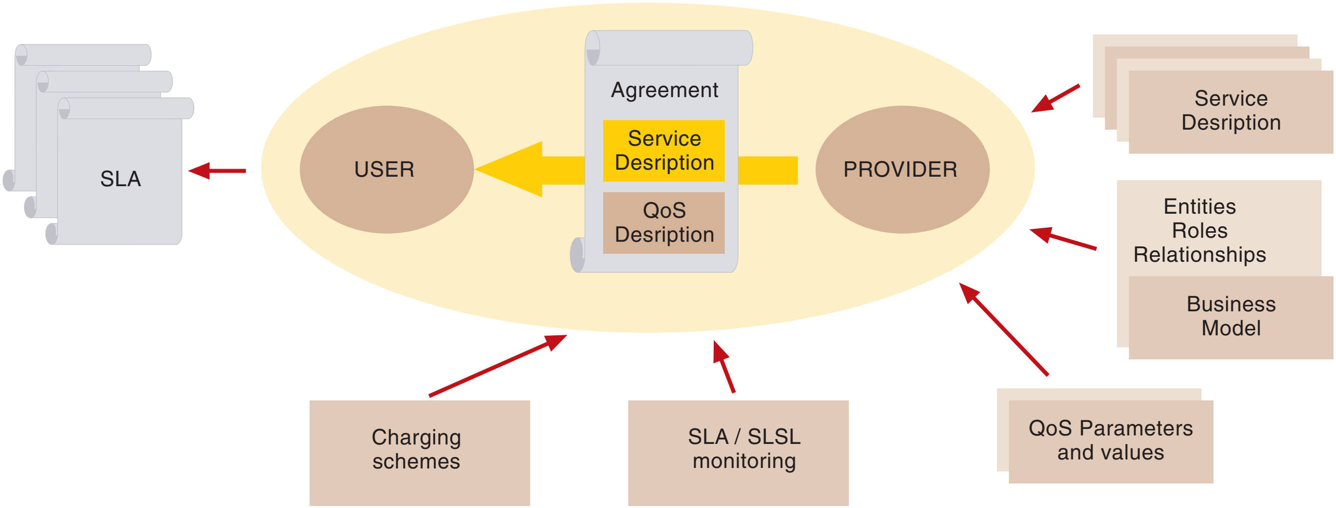


Figure 1 Handling SLAs – input needed by the provider results in SLAs

nents need to be bought from sub-providers in order to satisfy users' demands. As an output of running the procedure, the complete business model will be known (all the partners and the customers' segments will be decided upon) as well as the content of SLAs, e.g. QoS objectives, reaction pattern, etc.

This paper addresses issues related to the SLAs and SLSs. First of all, in Chapter 2, the relationships between different types of agreements present in today's IP-aware telecom market are discussed. In Chapter 3, basics of SLA, its types, structure and applicability are presented from a generic perspective. A suggestion for the generic structure of the QoS-related part of a technical portion of an SLA is presented, offering a possibility for providers to reuse the same structure each time the situation changes for them (either in the business or technical sense). The generic principles are elaborated further, specifically SLAs related to IP, as presented in Chapter 4. Some examples of SLAs offered in today's market for various IP-based services are presented in Chapter 5, followed by the status of the work undertaken in various standardisation bodies given in Chapter 6. A discussion on the mapping between SLAs and SLSs is conducted in Chapter 7. Finally, the paper concludes by analysing the future and possible evolution paths for SLAs and SLSs in IP-based networks.

2 Agreements and specifications – BLA, SLA, SLS, TCA

Various types of agreements/specifications that are discussed nowadays are depicted in Figure 2. These are all referring to relations between pairs of actors, pointing to the service provision configuration.

Generally, the agreement made between any user and any provider represents the harmonised understanding between these two parties by formally comprising/expressing the way they should behave. Their behaviour is described via a set of expressed duties, rights, and obligations.

Three types of agreements present in today's business and technical research – a Business Level Agreement (BLA), an SLA, and a Traffic Conditioning Agreement (TCA) – are described in this chapter. The original definitions made by the bodies/fora introducing these terms are quoted if available. In addition to the agreements

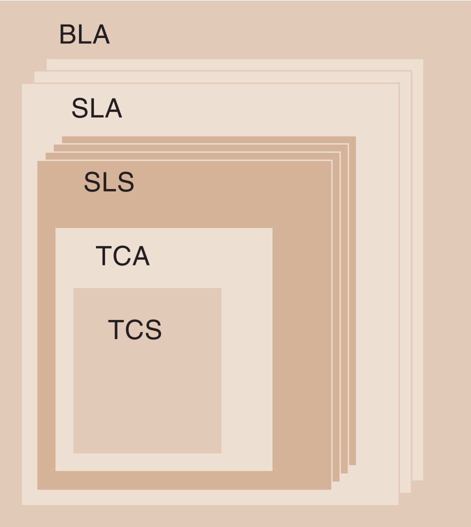


Figure 2 Illustrating relationships between BLA, SLA, SLS, TCA ans TCS[2]

[1] *Discussions on the procedure and trade-off/strategic decision undertaken by provider are given in [Tele0200].*

[2] *Note that any of the hierarchically superior agreements may contain a set of inferior agreements/ specifications. For example, a BLA may contain more than one SLA, e.g. if a service packet is subject to provision, or an SLA may contain multiple SLSs, etc. as illustrated in Figure 2.*

there are two types of so-called specifications included in the IP-based environment – an SLS, and a Traffic Conditioning Specification (TCS), which are also described here. The relationships between these agreements/specifications are not obvious, since different fora have made them for different purposes and from the perspective of offering different services on different infrastructures.

2.1 Business level – BLAs

On the business level between two actors[3] a Business Level Agreement (BLA) can be made. It refers to the agreement made between two legal entities/actors that includes a set of SLAs and reflects the global business relationship between the two actors involved. It is an 'umbrella' agreement, made on a business level, which defines the frame within which the partners may 'move' when negotiating any service to be provided/used between them. In this type of agreement, legal, economic (e.g. discount), regulatory, etc. issues are stressed, rather than technical details that are tackled in separate SLAs covered by the BLA.

BLAs are made for the provision/usage of a set of services, i.e. service packages. Such an agreement is actually a function of the SLAs made between the actors. For the generic case, the functional dependability between various SLAs (in particular the QoS parameters and their values) is not straightforward. It may be a mathematical function (e.g. a sum) or any relational function (e.g. lower than). On the other hand, it might also be a rather complex relationship between the sets of conditions. Assume, for example, that the delay is a relevant QoS parameter and its maximum value should be decided upon in the BLA. This value is a single value (which simplifies the problem slightly). Consider a selected set of traffic flows specifying their maximum delay requirements. Then, providing a single value (for these traffic flows), one may say that the worse case, that is, the minimum of the maximum delay requirements, should be included as the statement in the BLA.

Some simplification when considering sets of parameters may result in a less optimal use of resources. Some steps could be taken, however, like careful design, detailed specification of the services sharing resources, user profile capturing (e.g. time-of-day behaviour). Then, statements in the BLA may be refined and the improved usage of resources needed for assuring QoS used may be obtained. Therefore, understanding the dependencies between SLAs combined in a BLA helps making adequate strategic decisions and fulfilling user demands.

In addition, situations where congestion may occur are handled better, since reactions are stated in the agreement. The conditions agreed on in the BLA and SLA should be reflected in both network elements' configuration, mechanisms and management solutions.

2.2 Service level – SLAs

SLAs can be defined and used in the context of any industry in which a provider-user relationship exists. Hence, SLAs have been widely used in different industries and businesses, for outsourcing services, e.g. help desks, catering services, IT competence centre, etc.

For traditional telecommunication services (e.g. telephony) similar concepts covering similar aspects (e.g. QoS) have been applied, which might not have the form of an SLA. The presence and the concept of the SLA is rather unexplored for IP-based services, where many issues, both technological and business, have still to be studied. In addition, the situation is further complicated by the fact that there is a shorter time to roll out a service, the functionality included in the applied technology varies a lot, and the global picture of the market (user demands as well as provider roles) is steadily changing.

Basically, an SLA represents a harmonised understanding between a user and a provider regarding the service and the performance level of the service required from the provider by the user. It is designed in order to create and formalise a common understanding of the service, quality of that service, prices/pricing schemes, priorities, responsibilities, etc. In simple terms, it should specify what the user will get and what the provider is committed to provide. Various aspects of the relationships between the parties involved, like service/resource performance(s), help desk, billing, provisioning, service management, etc., can be included. As shown in Figure 2, an SLA would contain a set of SLSs (some sources restrict an SLA to containing a single SLS, but the issue of mapping SLAs and SLSs is not so straightforward as will be discussed in more detail in Chapter 7). SLAs contain much more information than only SLS(s), related to e.g. non-technical reactions, escalation schemes, legal, regulatory, economic, business, ethical issues.

[3] *Note that a provider taking a role in the market is here called actor. Any legal entity, e.g. a company, is an actor when appearing in the market and using and/or providing a service or service components.*

The DiffServ architecture [rfc2475] developed in the IETF defines an SLA as a "service contract that specifies the forwarding service a customer should receive". The SLA may include traffic conditioning rules which (at least in part) constitute a TCA. The DiffServ WG [diffserv] is changing their understanding of this term, since the terms SLS and TCS (see Sections 2.3 and 2.5) are introduced. The DiffServ WG came to believe that the notion of an 'agreement' implied considerations of a pricing, contractual or other business nature, as well as those that were strictly technical. There could also be other technical considerations in such an agreement (e.g. service availability) which are not addressed by DiffServ WG. Therefore the DiffServ WG agreed that new terminology would be used to describe those elements of service and traffic conditioning that are addressed by DiffServ. According to the latest draft on the DiffServ terminology [id-term], terms of SLS and TCS are introduced, as explained later.

Hence, SLA and TCA terms are considered in a broader sense than in [rfc2475], [rfc2474], [rfc2597], and this change will be introduced in the RFCs published by this WG.

SLAs are further discussed both in general and for IP in particular in Chapter 3.

2.3 Service/Service Component Technical Level – SLSs

An SLS consists of technical parameters and conditions related to serving a traffic flow using an IP transport service. An SLS would typically include:

- The type and nature of the service to be provided. All the components should be identified and described. The description of components related to particular interfaces is not a trivial task.

- The QoS of the service provided (this part will be elaborated on in the following section).

- The process of monitoring service provision (and QoS), i.e. which statistics to collect and present.

- Any technical consequences and the reaction pattern for the cases when either the user or the provider did not obey conditions agreed.

Additionally, the constraints on user behaviour may be included (e.g. the type of equipment necessary to experience the quality as agreed). Escape clauses may be included to define when the statements from the agreement do not apply – e.g. a fire damaged the provider's equipment, etc.

Note that SLS is a rather new term, introduced in 1999 by the IST project TEQUILA, and the topic is still under development. The definition given at the beginning of this section is adapted from [id-term] where DiffServ WG suggests the introduction of the SLS term. Their understanding of an SLS is that it is "a set of parameters and their values which together define the service offered to a traffic stream by a DS domain". The definition of 'Traffic stream' is unchanged from [rfc2475], that is "an administratively significant set of one or more microflows which traverse a path segment. A traffic stream may consist of the set of active microflows which are selected by a particular classifier." Simply, it means that a traffic stream can be an individual microflow or a group of microflows (i.e. in a source or destination DS domain), or it can be a Behaviour Aggregate (BA). Thus, an SLS may apply in the source or destination DS domain to a single microflow or group of microflows, as well as to a BA in any DS domain.

The results available in the IETF and in TEQUILA and other projects adopting SLS notions are presented in Chapter 6.

2.4 Traffic level – TCAs

This term is related to the provisioning of IP services and is defined by the IETF. According to [rfc2475] TCA is defined as "an agreement specifying classifier rules and any corresponding traffic profiles and metering, marking, discarding and/or shaping rules which are to apply to the traffic streams selected by the classifier. A TCA encompasses all of the traffic conditioning rules explicitly specified within a SLA along with all of the rules implicit from the relevant service requirements and/or from a DS domain's service provisioning policy."

Note that the TCA refers to rules executed at the border of a DiffServ domain. Thus a TCA is given for a certain class, identified according to the fields relevant for DiffServ. Examples of these sets for fields are found in [rfc2475] like:

i) Multi-Field (MF): combination of one or more header fields, such as source address, destination address, DS field, protocol ID, source port and destination port numbers, and other information such as incoming interface, or,

ii) Behaviour Aggregate (BA): based on the DS codepoint only.

TCA refers to a traffic flow, its characteristics and mechanisms to use in order to ensure/enforce that the characteristics are followed.

2.5 Determining traffic level – TCS

This term also relates to the provisioning of IP services and is defined by the IETF. According to the new terminology for the DiffServ WG of the IETF [id-term], a TCS is a term related to the DiffServ architecture, and should be understood as "a set of parameters and their values which together specify a set of classifier rules and a traffic profile". A TCS is an integral element of an SLS.

3 Service Level Agreement (SLA)

Facing the situations described in the introduction where changes are rather dynamic, the need to describe principles for efficient arranging of relationships between the actors is steadily getting more pronounced. Generally speaking, any relationship between two actors is associated with a set of expectations as well as a set of obligations. A *Service Level Agreement* (SLA) is an explicit statement of the expectations and obligations that are agreed between two actors: a customer and a provider [Verm99]. This term has been used for a long time, and therefore many different definitions of the term SLA exist. These are developed by different fora for particular purposes, and are basically not competing or overlapping, but rather have different focus. The discussion on the term itself is omitted here; some definitions of SLA (and more generally of an agreement) can be found in [NMF701], [P806d1], [Cain97], [Gray00].

Generally speaking, SLAs can be defined and used in the context of any industry in which a customer-provider relationship exists. Hence, SLAs have been widely used in different industries and businesses for outsourcing services, e.g. help desks, catering services, IT competence centre, etc. For traditional telecommunication services (e.g. telephony) similar concepts covering similar aspects (e.g. QoS) have been applied, which did not have the form and name of an SLA. The presence and the concept of the SLA are getting revitalised as an area of research in the IP-based environment, as discussed in the next chapter. The SLA is designed in order to create a common understanding of the service, quality of that service, prices/pricing schemes, priorities, responsibilities, etc.

A harmonised understanding may require a negotiation process, a result of which is (a set of) SLAs. The SLA negotiation process is very demanding, and the team of experts from different fields (e.g. technology (network architecture, QoS, management, billing), economy and business (strategic decisions, pricing, charging schemes), law (jurisdictional, regulatory issues), social science (anthropological, ethical issues) may be involved. Before even starting the negotiation process needs, gains, and business goals, should be determined from both provider's and user's point of view. Each team has to be aware of main goals, limitations, capabilities, which services to produce and which to buy ('home-made' vs. 'imported' services), and to have clearly determined core business and strategic streamlines. After that, before making an SLA, some preparation should be done so that the starting situation is determined and described. From a provider's point of view, operational capabilities and strategic position for different functionality/systems have to be addressed and known, e.g. billing, connectivity services, ordering/provisioning, network/service management, liability, usage, repair, collocation, performance reporting, customer relationships management, etc. In addition, the costs of performing each of the relevant functions should be analysed and compared with the costs of buying services from a sub-provider and agreeing the SLA with him. From the user's perspective the information of this difference in costs may build a basis to decide whether to go for a standard 'menu' solution offered by the provider, or to ask for a more specific 'tailored' solution for the SLA. In the latter case, the provider's team would usually adjust the terms according to the needs and wishes of the user. The responsibilities in the team can be redistributed according to the topics/issues to be detailed. After running the negotiation process and agreeing the SLA, SLA assurance (that should be agreed upon as well) has to verify that both sides keep the statements in the SLA.

Different practise and recommendations can be found for the SLA negotiation, and some guidelines of establishing and conducting a negotiation team are available, but this is not our focus here. As a result of negotiations, an SLA is made and would typically include:

- The **description of the service** to be provided. The description may be composed as a set of service component descriptions (i.e. service specification), or as a description of the service scenarios relevant for the user. Also, **a type and a nature of the service** to be provided. When describing a service, all service components should be identified and described on each of the interfaces between a provider and a user, which is not a trivial task.

- The **QoS-related part** handling the quality level of the service, including QoS parameter definitions and objectives. This part of the SLA will be elaborated further in the following section.

- The process of **reporting problems and troubleshooting**, which may include information of the triggering events, the person to be contacted if the problem occurs, the format of the complaint, the step-by-step process for troubleshooting, etc. The time period for resolving the problems should also be defined. Usually, an escalation matrix[4)] should be agreed on as well.

- The process of **monitoring and reporting** the performance and quality delivered. Here, the issues of measurements, which type of statistics, how often, where the measurements should be undertaken, the data collection, analysis, access to past statistics, etc. would usually be described. More details on this process will be given later in relation with the QoS part of an agreement.

- The **consequences and the reaction pattern** for the cases when either the user or the provider did not obey what was agreed in the SLA. Additionally, the constraints on the user behaviour may be included (e.g. for a telecom service it may imply the request for adequate type of equipment, for example a PC with given characteristics thay are necessary to experience the quality as agreed). Escape clauses may be included to define when the statements from the agreement do not apply – e.g. a fire damaged the provider's equipment, etc.

- **Legal issues**, which include the legal identification of the parties involved, responsible persons who are members of the SLA team, terms under which the SLA is not valid, when is it broken, etc.

- **Economic issues**, which may include tariffing policy, prices, charging schemes to be applied, penalties to be paid in case any of the events triggering the reaction pattern are detected, etc.

- **Regulatory issues** that may be extremely important and may include references to the directives restricting further retail of the service contracted, etc.

- **Other issues** that may include specific anthropological, ethical, ethnic issues that are of specific relevance for a customer or provider.

Handling SLAs and their negotiations is simplified if they have a generic structure, i.e. a template that can be reused for any service, business case and technology a provider might be dealing with. Summarising – an SLA should simply specify what the user will get and what the provider is committed to provide. Different types of SLAs are made and negotiated for different services, as discussed in the next section.

3.1 Generic SLA types

Various types of SLAs can be recognised having different aspects/parameters in focus. For example should be noted the differences in content and format of information relevant to different users and providers.

Regarding the content, an SLA can be general/universal and made on a 'one-suits-all' strategy when offered to e.g. a large segment of residential customers, or it can be more specific, adjusted specifically to customer needs, i.e. 'customer-tailored', suitable for a particular business customer. Regarding the details included in descriptions/statements given in the SLA, in case the SLA is made between the provider and the residential user, the granularity of parameters would be chosen naturally so it fits a large number of customers. That means that the selection of parameters will be easy to understand, and should not be expressed strictly technically.

Even the language used to describe for example QoS issues/parameters would be less technical and understandable to the actual user. On the other hand, an agreement between two providers would be more complex (e.g. include more parameters) and expressed in more technical terms. For example, the end-user can understand that its service will be unavailable for less than 5 minutes per month, which actually in technical wording is equal to an availability of 99.99 %.

Regarding the dynamics in the SLA negotiation and contracting period, commonly for outsourcing services in industries other than telecom (e.g. catering, etc.) a negotiation period runs from six weeks to three months, depending on the scope and volume of the contract, while a contracting period runs for 3–10 years. In telecommunications, the dynamics of SLAs is more pronounced, since an SLA may be contracted for different time scales. The granularity of time varies from monthly/yearly (e.g. telephony service subscription, monthly subscription to the Internet Service Provider (ISP) for the Internet access service, renting out a fibre) up to very short periods like 10 minutes, or per session (e.g. one transaction for e-shopping, accessing ftp server, downloading certain content, etc.). The

[4)] *An escalation matrix indicates the hierarchy/degree of importance of problems, and the information necessary to solve the problems. This information may include e.g. persons to be contacted, alarm description, messages to send out, format of messages, etc.*

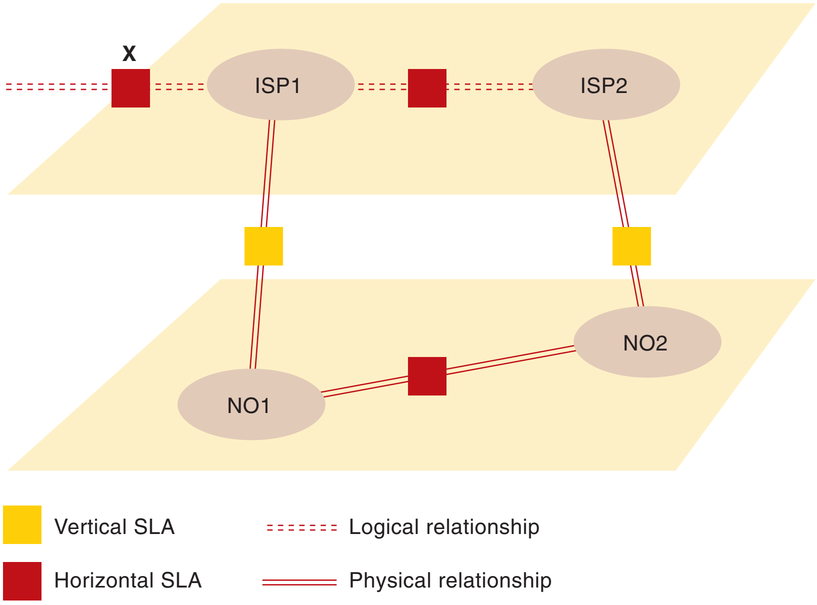


Figure 3 A simple example of relationships in multi-provider Internet environment

dynamics cannot be realised if the mechanisms for negotiation and management are not in place. Some work is done in the Internet2 project on bandwidth brokers and dynamic SLA negotiation [I2-site].

Depending on the interface an SLA is related to, it can be either vertical – between two actors on different layers, or horizontal – between two actors on the same layer. One example is a configuration as shown in Figure 3, where ISP1 has to rely upon both ISP2 (horizontal relationship formalised in the horizontal SLA) and upon the Network Operator 1 (NO1) (vertical relationships formalised in the vertical SLA) in order to fulfil the demands of its user served via interface X. Note that depending on the user tye, the SLA at the interface X may be either vertical (e.g. human end-user) or horizontal (e.g. another provider).

Depending on the logical location of the parties involved in the SLA, it can be:

- Internal – made between two different departments/business units within a company;
- External – made between two different legal actors, i.e. two companies.

Also, depending on the performance level (of the service offered) handled in the SLA, three main categories of SLAs are recognised (Figure 4):

1. *Application-level SLA* – covers the service end-to-end, i.e. including not only the network infrastructure edge-to-edge but application(s) and Customer Premises Equipment (CPE), implying that an actor playing the role of network operator only, cannot provide such an SLA since it does not have control over either

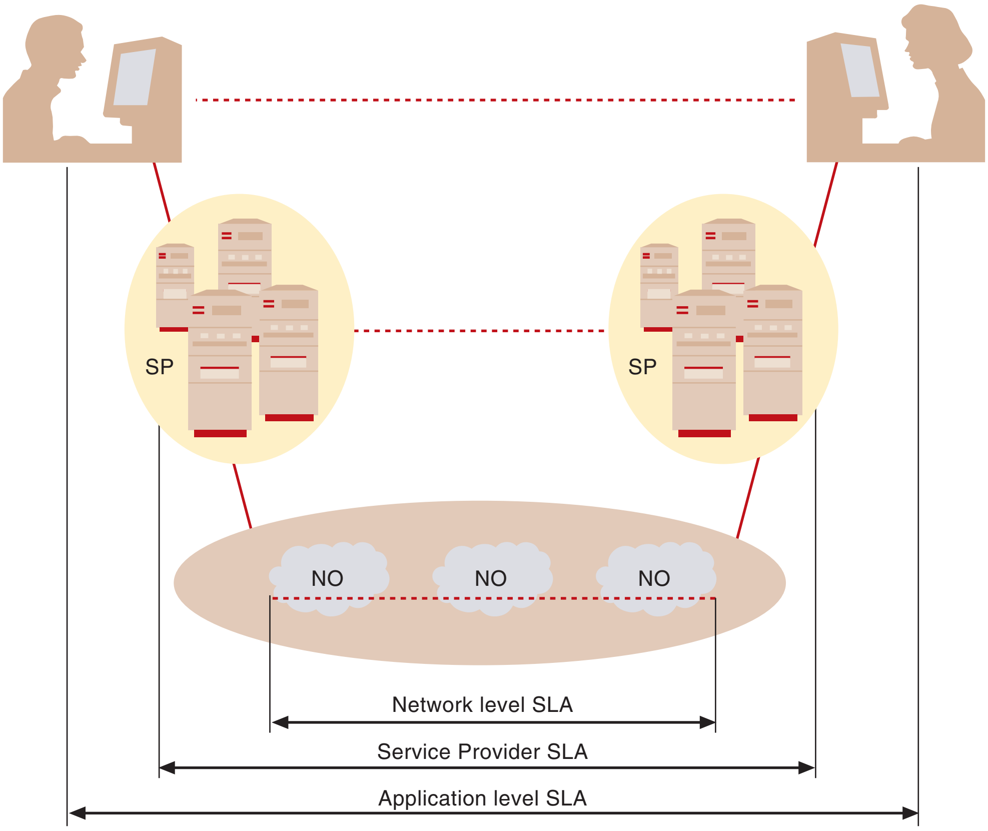


Figure 4 Illustrating different types of SLAs depending on the scope of the service provided

the end-hosts or a Service Provider (SP). The statements handling the performance and the service level should be expressed in terms of application units that the user understands and is concerned about, e.g. the time to complete the transaction instead of round trip delay and some additional information. This type of agreement also includes the constraints on the user and his behaviour and minimal requirements placed on the equipment, since the application using network services is known. It is a common practice today that the characteristics of the customer owned/administered equipment (i.e. user's network) are described by using agreed parameters. When considering the QoS-related part, a selection of QoS parameters must be devised by focusing on the application, e.g. a parameter of availability of 99.995 % of time may be expressed as the only 25 minutes per year (3 minutes per month) that the service will be unavailable. The values of the relevant QoS parameters are determined by taking into account the peculiarities of a particular implementation of the service.

2. *Network-level SLA* – specifies statements in terms of performance observed while providing a network connectivity / transport service. This agreement is usually made between two providers, though in case a customer is e.g. a large company it may be made between an end-customer and a provider agreeing upon the usage/provision of a transport service. One type of the network level SLA is a peer-to-peer agreement made between ISPs. The parameters used to describe the performance of the network and the quality of the network service(s) are very detailed, technical parameters whose objectives may be described by using different statistics and moments. Depending on the scope of the service, and its implementation various types of network level SLAs exist. For example, a Leased Line (LL) service may be supported by the infrastructure that is Asynchronous Transfer Mode (ATM) based or Frame Relay (FR) based, for which ATM- and FR-related parameters are used, respectively. The network level SLAs for IP transport services are elaborated on in Chapter 4.

3. *Service provider SLA* – used by providers offering server-hosting capabilities. The provider (usually an actor playing a role of a service provider, or a co-location provider) has control over the server side but not over the customer side or the network performance. In this case, parameters related to the performance of (various types of) servers, or other (peripheral) equipment are relevant. The parameters may include the performance of a database, e.g. a number of simultaneous transactions that a database can serve, a number of simultaneous hits on the web-content that a web server can support, etc.

3.2 Generic Structure of the QoS Part in an SLA

In order to handle both the increasing volume of SLAs, their complexity and to assure their maintenance, having the generic structure/template that can always be (re)used would help a lot. Being *generic* implies the structure's independence of service type, network, technology involved in service provisioning, type and organisation of actors involved, etc. On the other hand, being generic does not exclude considera-

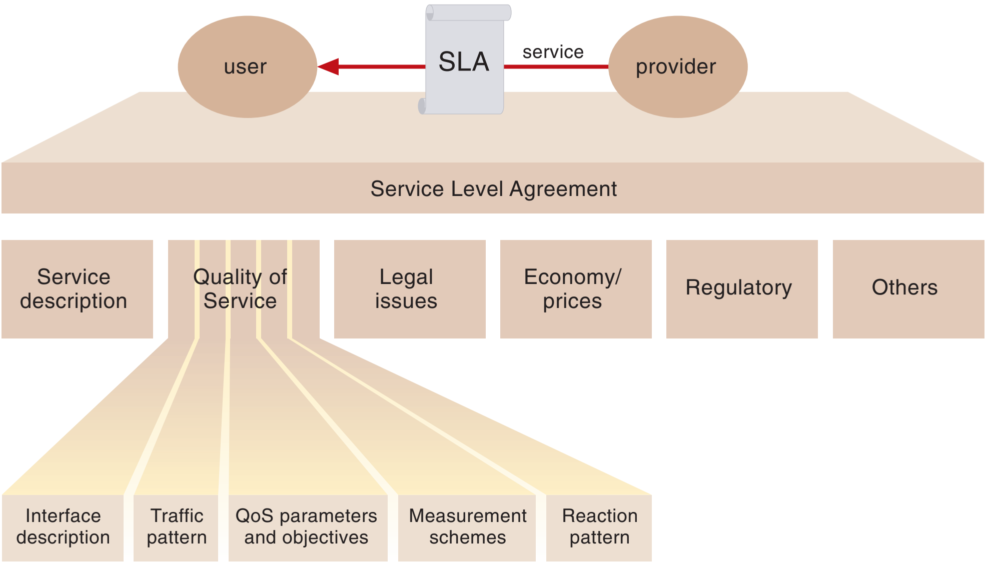


Figure 5 A structure of an SLA – focus on the QoS part

tion of specific situations at each interface. That implies that a set of mandatory statements should be available; these can be generally applied in addition to a set of optional, service specific and/or user-profile specific statements.

Figure 5 illustrates a provider and a user where the provider delivers the service to the user which makes use of that service. The SLA between them includes the QoS-related part, which is focused on in the following. More detailed description of the QoS-related part of an agreement can be found in [P806d1]. The illustration of the structure of the QoS-part of an SLA is given in Figure 5.

Interface description includes the description of all the interaction points relevant for the agreement – both business and technical. It might contain the information on the service delivery point, protocol(s) to be used, measurement points, observation points, points where a reaction pattern will be applied, negotiation points, etc.

Traffic pattern description describes the characteristics of the expected traffic flows. This information allows the provider to manage resources in its domain in order to deliver the agreed QoS. The description of the traffic should envelop both application and management information flows. The characteristics of both the ingress and egress traffic should be described. Traffic patterns can be described on different time scales (e.g. during the day, per service instance, etc.). The parameters used to describe the traffic could for example be average or higher order moments.

The description of **QoS parameters and objectives** implies expressing the performance of a service by assigning values to a number of QoS parameters [ETR003]. The QoS parameters can be derived by applying the adapted ITU-T 3x3 matrix [I.350]. Considering QoS objectives, they can be specified by target values (e.g. total maximum delay), or by thresholds set to a QoS parameter, e.g. an upper (or a lower) bound (e.g. an upper bound for unavailability). The QoS objectives may also be expressed as guarantees – provider's commitment to the user with strict traffic and reaction patterns, or as QoS indications, which are associated with loose traffic patterns and slow reaction patterns. Since QoS objectives are closely related to both measurements and reaction patterns, both measurement procedures and conformance rules should (e.g. statistically) fit the granularity set to the QoS objective.

The **measurement schemes** description should include the statements who, where, when, and how should measurement and conformance testing processes be performed for the agreed parameters. The description may include: the identification of relevant measurement points, the specification of the measurement environment, description of the technique(s) for obtaining the measured values, specification of the methodology to present and evaluate the results by parameters, and the method to be used for taking decisions on acceptance based on the level of compliance of the measurement results with the stated requirements and commitments.

A set of **reaction patterns**, related to failure to meet either traffic patterns or one/more of the agreed QoS objectives should be described in the QoS agreement. Such a description may include the reaction patterns both for cases of detecting the non-conformant traffic and for detecting the QoS degradation. The reaction patterns for both entities should be stated including the inputs to initiate the reaction (e.g. results of measurements), related constraints (the duration, timeliness, type of actions), resources and tools required to carry out the reaction, and the description of the reaction itself. The reactions could be technical (policing the traffic flows, suspending or aborting the activity, sending alarms, warnings, etc.), economic (e.g. discounts, initiation of using compensation schemes), legal and ethical (e.g. publishing the "antispam black lists"), etc.

Though the description of several terms (like traffic flows) is here commonly related to the operational phase of telecommunication services, this could be generalised in order to be applicable for every service life cycle phase. The corresponding terms should then be adapted in order to describe better the relevant aspects.

4 SLA for IP-based Services

The area of SLAs is revitalised with the challenges faced by providers offering services with assured QoS in an IP-based environment. Different aspects of SLAs, their content and the selection of QoS parameters and their values are still under development when it comes to IP services. The reasons are multiple: dynamic changes in the market offering multiple services on a single infrastructure that is based on the IP technology that is still not mature (though many solutions, mechanisms and architectures are developed for supporting QoS in such an infrastructure, there are still many effects that are not fully explored); rapid changes in business models where multiple providers are offering similar services often aiming at the same market segments; the applications developed lately ask for higher quality than the applications traditionally developed for best-effort IP networks; users are having higher demands being advanced by a simple-to-use application based on IP. The situation is becoming more complicated by a shorter rollout time for services, the functionality included in the technology used nowadays still varies very much (i.e. different technologies are still present in the infrastructure supporting services like data transfer, video, telephony, etc.), the technologies present in the access portion are still evolving towards those who can support high demand real time services. In addition, the roles of a user/customer and a provider are not so clear like in the traditional telecom market, since any user may be a provider to his users without having an extensive set of equipment, but some value added to the basic transport service(s). In short, many issues, both technological and business, have still to be studied in order to enable providers (and users) to capture and react in the global IP market.

Understanding SLAs and their handling is important for any provider that has to rely upon its partners to offer any global service, and that has to fulfil requirements of its users that may easily use another provider if not satisfied. Having a generic structure as presented in the previous chapter, enables providers to react fast (even automated) when adapting to changes such as a new role, introduction of a new service, changes in the existing offer, introducing additional mechanisms in the infrastructure, and so on. Also, a process of negotiating SLAs is getting more structured when using a template that is independent of services, technologies, business. Another important issue is that the practice used in the traditional telecom does not have to be abandoned by introducing new (multi-provider) concepts and using SLAs.

Historically, the first SLAs in the Internet were of peer-to-peer nature for public Network Access Points (NAPs), whereby large backbone providers make bilateral interconnection agreements where the main issue is stating that the volume of traffic a provider injects into the partner's network is equal to the traffic he should allow coming into his own network from the partner's network (i.e. from a peer to a peer).

4.1 SLA Types for IP-based Services

SLAs for IP-based services will naturally be of the same generic SLA types as presented in Section 3.1, i.e. SLAs will differ according to the level of detail included, language, nature of parties involved, contracting period.

The interesting issue is actually whether an SLA is agreed on a timely basis, i.e. in traditional manner for a time period of e.g. one month (like subscription for telephony service), or per session, i.e. per each instantiation of a service. The result of having SLA per session is a tremendous amount of SLAs (and their respective content) that have to be handled by providers. Therefore, the issue of SLA management is getting more pronounced.

Regarding the performance level handled in the SLA, three main categories are recognised:

1. *Application-level SLA* – typically includes statements for specifying the performance of a specific application, e.g. the response time of a database server will be less than 100 ms, with a maximum of 100 clients connected simultaneously, or the video server will be available 99.9 % of the time during the evening hours (between 1800 and 2400 hours), otherwise 95 % of the time. Another example is statements included for a VoIP service. The parameters chosen should reflect both the speech quality and the connection quality. In this case, different parameters are used to express the quality of speech, and others to express the connection quality. Some of these are: Mean Opinion Score (MOS) for the speech quality, and set-up delay for the connection quality. Also, the differences caused by e.g. realising a Voice over IP (VoIP) service by implementing ITU-T's H.323 [H.323] or IETF's Session Initiation Protocol (SIP) [rfc2543], [sip], should be taken into account when defining thresholds (or other objectives) for the chosen parameters. For example, an application level SLA for the VoIP service may include the following: (1) the H.323 protocol suite is implemented, (2) NetMeeting™ is used as a client, (3) CPE characteristics (including both hardware and software) are specified, e.g. a Personal Computer (PC) with a minimal hardware of a Pentium II (or equal) CPU, 64 MB RAM, sound card, phone card, loud speakers, microphone, or equivalent headset, access to the IP network, and a minimum software of the H.323 suite implemented, and NetMeeting™, etc.

2. *Network-level SLA* – deals with the performance of the network offering a transport service. Traditionally, in the Internet, a best-effort service is offered, but now the paradigm is changing in order to support real-time services. Three different approaches can be recognised[5] (Figure 6):

   - *Tunnel approach* – defines aggregate QoS between two specific points in the network.

[5] *Some argues that this way of organising SLAs better reflects the scope of the transport service provided, i.e. point-to-point (tunnel), point-to-multipoint (funnel), many-to-many point (cloud/hose).*

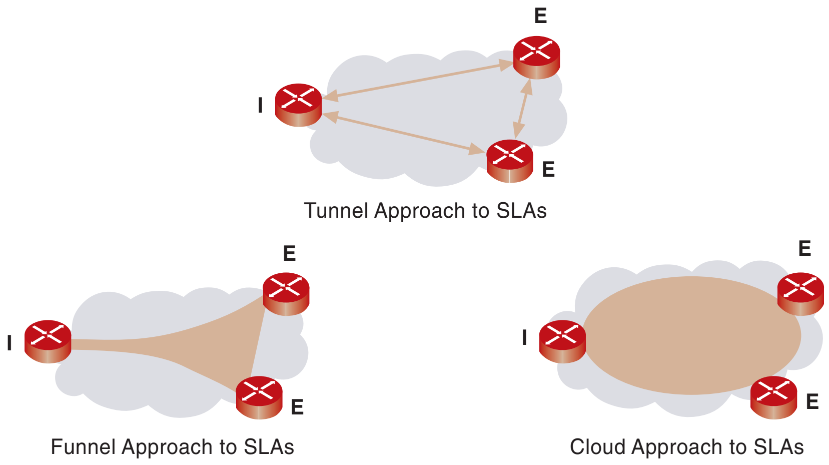


Figure 6 Types of SLAs for a network connectivity service

An example is the service for traffic transported e.g. from Telenor R&D at Kjeller to the Telenor headquarters in Oslo. This approach is depicted in Figure 6.a, where the points A, B and C are defined as input/outputs for a traffic stream.

- *Funnel approach* – defines aggregate QoS between a specific ingress point to multiple egress points. An example is the QoS for IP transport services from the VoD server to any IP-based user. This approach is depicted in Figure 6.b, where the ingress point goes to a range of egress points for a traffic stream.

- *Cloud approach* – defines aggregate QoS between multiple points in the network. An example can be IP-based transport services realised within one network (note that one network here implies the network controlled/owned by a single operator). This approach is depicted in Figure 6.c, where any of the points attached to the 'cloud' can be either ingress or egress point for a traffic stream.

3. *Service provider SLA* – this type of the SLAs are of special interest in the IP-based environment, since they open for many 3rd party providers, and a myriad of service providers may be present. The operator has control over the server side but not over the customer side or the network performance. An example of such an SLA can be relevant when designing the SLA for an Application Service Provider (ASP), or when offering web-hosting/co-location service.

4.2 QoS Part in SLA for IP-based Services

Following the generic structure presented in Chapter 3, the content of an SLA for an IP service may be made. Devising details is not possible if there are no 'clear' business model (i.e. roles and relationships), service description and implementation that all influence the final choice of QoS parameters, grouping into QoS classes, and the values of the parameters.

The issues of how the ITU-T's I.350 3x3 matrix approach may be applied for an IP transport service and how the parameters are devised, are discussed next. According to [I.350] QoS parameters could be primary or derived, and may be expressed in a more technical or more "humans understandable" language. Primary parameters are those that are necessary to measure to prove the performance as agreed, while derived parameters are actually functionally dependable on the primary parameters and do not need to be directly observed (i.e. measured/monitored).

Considering the primary QoS parameters, the 3x3 matrix approach identifies three generic functions:

- *Access*;
- *Information transfer*;
- *Disengagement*.

Also three criteria for characterising the realisation of the generic functions are defined:

- *Speed* characterises the temporal aspects of QoS associated with a function, showing time related efficiency characteristics, e.g. latency/delay.

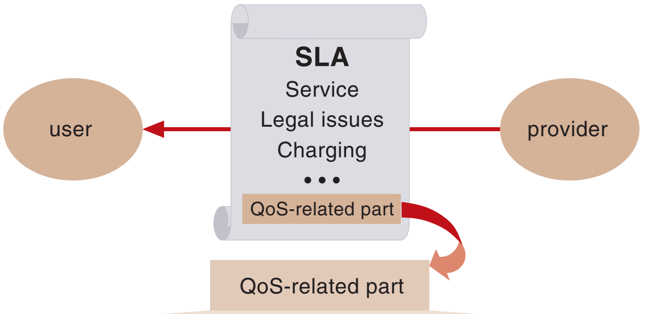
- *Accuracy* characterises the degree of correctness with which a given function is realised, e.g. ratio of errored packets.

- *Dependability* characterises the degree of certainty that a function is performed, e.g. ratio of failures on total attempts.

Although the above definitions for primary QoS parameters apply to telecommunication services, they can be generalised even for other types of services.

As mentioned before, derived QoS parameters are defined as functions of others. In particular, derived QoS parameters can be defined on the basis of observed values taken by primary QoS parameters, and of decision thresholds for each relevant primary QoS parameter. One example of derived QoS parameters could be those characterising the availability of a service.

When making a list of requirements the end-user may express the request in a form of non-technical parameters. Such issues should be taken into account by the provider and mapped into technical parameters related to those aspects.



Interface description Business interaction points Technical interaction points	Performance reporting point – web site URL SAP – IP address/port, phone number for dial up access Protocols – IPv4, BGP4, SNMP3	
Traffic pattern	Throughput 08-19h Throughput 19-08h Time of day pattern	384 kbps 128 kbps distribution function included
QoS parameters and objectives	Availability Loss ratio Set-up delay	99% < 5% < 2 sec
Measurement scheme	Measurement points Measurement methods Tools Granularity	Ingress router, egress router, gateway Active probing Ping, traceroute 10 minutes
Reaction pattern	Discount Traffic shaping Alarm Warning Policy invocation	Availability guarantee breached 1h UA 4h = 1 day service credit 4h UA 8h = 7 days service credit UA 8h = 30 days service credit IF user's traffic exceeds 2 Mbps at the ingress THEN discard it UA = unavailability

Figure 7 Generic structure of QoS-related part of an SLA, some examples

Some examples of the content of an SLA for an IP connectivity service are illustrated in Figure 7.

The interface description includes the description of the business interaction points – performance reporting point, service access points, and protocols/formats used for exchanging information on these protocols. Traffic pattern is basically described by the throughput which is linked to the time-of-day distribution and user's behaviour, where different values are allowed for different time periods. The most mentioned QoS parameters related to the provision of real-time services in the IP-based environment are delay, loss and jitter. Also, availability is proven to be a very important factor for customers. Hence, these are usually used in SLAs independently of the SLA type. In case of value added services, additional parameters related to the quality of customer care, e.g. help desk availability, mean time to repair, etc. are given as well.

Applying the 3x3 matrix to the case of a Virtual Leased Line (VLL), where the access and disengagement functions are not relevant, the QoS parameters identified for the IP packets transfer phase can be listed as:

IP packet transfer delay expressed both as a mean value and variation [Y.1540], [rfc2330], [rfc2123]	transfer speed
IP packet error ratio [Y.1540], [rfc2330], [rfc2123]	transfer accuracy
IP packet loss ratio [Y.1540]	transfer dependability
IP transfer availability [Y.1540]	transfer dependability

In order to prove delivery of the quality of the service as agreed in an SLA, the chosen parameters need to be observed whether they are conformant to the agreed values or not. Also, the behaviour of the users has to be observed, so the provider may react when the user does not obey the agreement (e.g. generating more traffic than allowed in the SLA). Therefore, measurement schemes have to be developed and agreed – active vs. passive measurement methods, frequency of measurements that are chosen so they are not too much of an overhead for the servers, etc. The relevant measurement points need to be identified, e.g. ingress and egress routers, gateways, etc. Different metrics may be agreed as well, e.g. IETF's IP Performance Metrics (IPPM) [rfc2330]. The results of monitoring and measurements can be input to the other parts of the system, e.g. charging and billing system by sending alarms for initiation of different charging schemes after the various situations occur, etc.

Reaction patterns cover technical actions like alarms, warnings, messages sent to the user that did not obey the agreed traffic pattern, or it can result in abandoning the service usage. It may also involve some traffic engineering actions, like traffic shaping, call admission control initiation after receiving a trigger from the monitoring system, policy initiation, etc. On the other hand, economic reactions are much more obvious to the user, and are expressed as e.g. discounts when the quality has not been delivered by the provider as it was agreed in the SLA. Different compensation schemes are often used in today's SLAs as will be shown in the next chapter.

5 SLAs for IP-based Services in Practice

It is easier to capture the importance of an SLA when a case study is examined. The examples below will illustrate how the structure described before can be used in practice. The elements of the QoS-related part of an SLA can be identified in a more or less similar form in all examples illustrated in this chapter. Note that the actual content may differ, e.g. selection of parameters, values, statistics, but the structure is not diverging. This section aims at highlighting the fact that the structure could be generic and standardised, while the content may differ for a particular service, interface observed, provider involved, etc.

Examples described here include SLAs for some of the services offered by UUNet and Epoch Internet™.

5.1 UUNet's SLAs

Businesses realised the advantages of the IP technology, but they want their services to be guaranteed to them. For example, they want fast, reliable and robust Internet access. UUNet [uunet], a Worldcom company, was one of the first ISPs understanding the need to offer guarantees in the form of an SLA to its customers. UUNet offers a wide range of services. A range of services provided may vary depending on the country in focus.

- Internet access, including dial-up and remote access (suitable for single users, SME, or companies with many remote workers, i.e. with a need for home-office solution), dedicated access over DSL, ATM, FR for companies in the USA which have higher needs than remote access, and LL dedicated access;

- Hosting/co-location services;

- IP VPN and Internet security solutions;

- Internet multicasting;

- Wholesale services for ISPs and carriers (used by AOL, CompuServe, etc.).

The services offered by UUNet are rated as rather popular, as shown for example in the fact that 78 % of Times 'Top 100' companies and 63 % of Times 'Top 1000' companies use UUNet [top100]. UUNet offers SLAs for the Internet access service that are universal in structure but figures may deviate for each of the countries UUNet is operating in.

A brief summary of the SLA for Internet access service would include the fact they are network connectivity SLAs of the cloud type. This type is used since it supports similar applications across different access points, and is the easiest to specify, track and manage. UUNet monitors the network continuously to ensure that all the metrics defined in SLAs are satisfied. UUNet does not specify strict delay bounds, but instead provides an average delay. The SLAs [uunet1] include guarantees on:

Network quality including:

- *delay* ~ averaged monthly ≤ 65/85/120 ms, for US/European/Trans-Atlantic networks, respectively, measured as RTD by using ICMP[6] echo messages;

- *packet delivery rate* = monthly averaged ≥ 99 % for US, EU, Trans-Atlantic links, service quality (network availability of 100 % of the time, apart from scheduled maintenance in time windows agreed with the customer).

Service quality including 100 % availability for the network, e.g. for the VPN service UUNet uses *ping* to monitor the customer's router every two minutes to determine network availability. If the router does not respond after three pings, UUNet will deem the service unavailable and let the customer know immediately. Failure to do so will result in the customer's account being credited for that day. The estimated number of lost ping packets yields information about the avail-

[6] *ICMP measurements will replace the current NTP measurements (run each 15 minutes) as the technology used to gather the data on delay.*

ability of network connectivity. The scheduled maintenance is not counted as outage; it will be announced to the customer at least 48 hours in advance, and it may be scheduled for regular window periods (e.g. from 7 a.m. – 9 a.m. every Tuesday and Friday). Network failure will result in the customer's account being credited for each hour of down time. UUNet will notify customers within 15 minutes in the event of network failure.

Customer care quality – UUNet will contact the customer after detecting the unavailability on his links via pager, phone, fax or e-mail. The period to notify the customer of the outage is specified in SLAs, but varies for different services / market segments. Failing to notify the customer in the agreed period causes the possibility to realise UUNet's crediting of the service. Regarding *installation*, the time given in SLAs varies from service to service, e.g. UUNet guarantees that the installation of the circuit and the activation of the UUNet port will be completed within 20 working days for 64 kbit/s leased line services and 40 working days for 128 kbit/s to 2 Mbit/s services. For 8 Mbit/s services, installation and activation will be completed by the date provided in writing by a UUNet sales manager. Failure to do so will result in the customer receiving a 50 % refund on the start-up charge.

The reaction pattern in case any guarantees are broken by UUNet results in service crediting. The customer has to initiate a remedy procedure in the agreed period (e.g. 5 days from outage for reporting it to the local UUNet representative, which is a contact person stated in the SLA). On the other hand, for some services (e.g. IP VPN) constraints are put on the amount of traffic the customer can inject into the network (if it exceeds 50 % then the customer has to initiate the request for capacity upgrade within 30 days) [uunet2].

5.2 EPOCH Internet

Epoch Internet [epoch] is an ISP that provides high-performance dedicated access and VPN and web hosting/co-location services to customers in the USA. Epoch Internet offers SLAs for its service for dedicated access [epoch-a] and web-hosting [epoch-w], which have basically same structure, but are service specific for certain parameters, and the values of parameters differ. The SLA for web-hosting/co-location services [epoch-sla] is an example of an SP SLA for hosting several servers on behalf of its customers. It is typical for these SLAs to use uptime and the performance of servers as a gauge for SLA satisfaction. The crucial issue for this type of SLA is to assure that presence of multiple users (since they host multiple customers) simultaneously does not affect the SLA performance guarantees.

Verification of SLAs is done by continuous monitoring, at the time granularity of 10 minutes. Network outage notification and availability is one of the guarantees included in this SLA. To warrant network reliability, the SLA contains a Packet Loss guarantee (monthly average packet loss < 5 %, measured each 10 minutes, averaged monthly) and an Internet Latency Guarantee (monthly average < 85 ms, measured each 10 minutes, RTT on the Epoch network, averaged per month). In case any of the guarantees are broken by Epoch, a scheme for claiming service credits should be supplied by the customer. If confirmed by examining monitoring data, the customer will receive the discount. A service credit in the SLA implies that the service is granted for the period the service credit is given. Epoch bases its pricing policy on the service credit issued upon establishment and approval of a service request. The customer will be billed monthly and failure to comply with Epoch's terms and conditions will cause a breach of the SLA. This SLA is created on the 'one fits all' basis and Epoch may at any time choose to change, amend or revise its conditions by posting it on its website. Regarding availability the following is guaranteed for webhosting/co-location services[7]:

Hardware availability – 99.9 % for all the hardware sited in the Epoch network;

Power availability – 99.9 % for supply of AC power to the components in case of co-location service;

Core applications availability – 99.9 % per month for all services, but not for ROOT service[8];

Data centre availability – 99.9 % from customer to data centre for co-location customer, and only to data centre for web-hosting customer.

Backbone Network Availability Guarantee – Epoch guarantees that the network will be available 99.9 % of the time (excluding any planned/scheduled downtime that Epoch informed the customer of). Any customer who has experienced unavailability (UA) of links, may claim service credits according to the following rules:

- 40' ≤ UA ≤ 4h implies service credit of 1*1/30 (one day);

[7] *For the dedicated access services only the backbone network availability applies.*

[8] *For detailed description of the Epoch services visit [epoch].*

- 4h ≤ UA ≤ 8h implies service credit of 7*1/30 (one week);

- UA ≥ 8h implies service credit of 30*1/30 (one month).

Other guarantees include:
Outage Notification Guarantee – Epoch guarantees the customer that it will inform the customer of any unexpected network unavailability within 1h of the occurrence of any guarantee break. The customer will be informed by telephone or email. If notification is delayed or not provided, service credit of one day applies.

Internet Latency Guarantee – An average monthly transmission rate of 85 ms or less is guaranteed on the Epoch network. Latency (average round trip transmission) is measured at 10-minute intervals and the average is calculated monthly (at the end of every calendar month). Service credit of one week will be provided if the average Internet latency for a customer is greater than 85 ms for any calendar month. If it happens in two successive months, the service credit is one month.

Packet Loss Guarantee – The average packet loss ratio on the network will not be more than 5 % during any calendar month. It is measured every 10 minutes and the average is calculated at the end of each calendar month. One day of service credit will be given to customers who experience a packet loss higher than 5 % per calendar month.

Installation Guarantee – for web-hosting/co-location services is implied 21 days after the request is submitted. For a dedicated access service, the installation requires up to 38 working days after an order has been accepted and entered into Epoch's provisioning system. If this is not met, the customer will receive one month of service credit provided that the delay was not caused by the customer or by any Force Majeure. The equipment must be provided or approved by Epoch and the customer has to co-operate, e.g. to be present during the installation.

Regarding claims for service credit, they must be submitted by a customer within seven business days of the end of the month during which the event occurred that gave rise to the claim. Epoch also reserves the right to make changes to the service level agreement.

6 International Fora and SLAs/SLSs

As mentioned before, SLAs are present in the telecom market, but in a slightly different form. With emerging challenges for providing differentiated IP-based services with QoS guarantees, the interest for SLAs has increased rapidly. Hence, the research on this topic is intensified. In this section some examples of the work and results from various standardisation bodies and international fora related to SLAs are given. First, an example of ITU-T agreement for the telephony service is shown, followed by some examples and the ongoing work from IETF on the SLS topic, which is basically initiated by IST projects Tequila and Aquila. The example of so-called end-to-end SLA, from TMF (ex. NMF) is described afterwards, and the EURESCOM projects P806-GI and P906-GI understanding of SLAs and related issues are given at the end. Note that several bodies studying this topic are not discussed here, e.g. the DTMF SLA group.

6.1 International Telecommunication Union (ITU-T)

One example of an agreement can be found in ITU-T Recommendation E.801 [E.801], which describes a so-called Service Quality Agreement (SQA) that is defined as *"a bi- or multi-lateral agreement between interconnecting ROAs*[9]*, network providers and/or service providers, to initiate a formalised programme for the monitoring, measurement and setting of targets intended to satisfy the end-user and other customers. When appropriate, mutually agreed action plans will be developed to improve a target that is below the expected level of performance"*.

An SQA includes the following:
- Introduction, e.g. describing the purpose of the agreement;

- Scope, e.g. the services covered and corresponding interfaces are to be presented;

- Confidentiality, for instance stating the confidentiality concerning the content of the agreement and sharing information between the user and the provider;

- Legal status, like stating the commitment to fulfil the conditions;

- Traffic patterns, e.g. describing relevant characteristics of the traffic flows;

- The relevant QoS parameters and corresponding (range of) target values;

[9] *ROAs (Recognised Operating Agencies).*

- Measurement schemes, like describing points of observation, events to register and ways of aggregating the data;

- Reaction patterns, e.g. describing ways to act in case any of the conditions are not fulfilled;

- Management review process, like presenting procedure for reviewing the agreement; and

- Signatories.

This template is traditionally used for agreeing on the provision of e.g. telephony service between different Public Network Operators (PNOs).

6.2 Internet Engineering Task Force (IETF)

The presence and the effect of SLAs in the IP world should be discussed bearing in mind both existing best-effort services and the IntServ [rfc1633] / DiffServ [rfc2475] architectures suggested by IETF.

The best-effort service implies no guarantees for the service provision, and hence asks for no agreements in the operational phase. One argument is that available resources are shared indiscriminately between the users; one service for all. Some explicit statements describing the volume of traffic to be exchanged, e.g. between two peering ISPs, can be found. On the other hand, when DiffServ/IntServ-RSVP architectures are implemented, the SLA becomes more pronounced as a way of regulating the conditions for service provisioning between a customer and a provider, e.g. between an enterprise using the services provided by an ISP.

Work currently under development within IETF [policy], allows for a description of SLA schemes in an abstract common language. SLA schemas are described by a set of attributes. The attributes may either be common to both IntServ/DiffServ or specific for each of them. The common attributes can include name, scope, type, address range and max rate. Specific attributes for DiffServ are e.g. Type Of Service (TOS) field masks and patterns, and for IntServ e.g. flow service type, maximum flows, token bucket parameters, etc. On the other hand, SLA can be structured by use of references. This allows the definition of generic service profiles like a premium, gold, standard service package, or a generic customer class profile like economy, professional, etc. IETF has focused more on TCS/TCA as described in Chapter 2, and lately a lot of work is done on this topic. As already mentioned SLS describes the technical details related to the level of the service provided to a traffic stream. It is a rather new term, established in the IST Tequila project, which at the moment is trying to consolidate interests on this topic and start a WG in IETF. Four IETF drafts have been published on this topic, and a number of RFCs related to the DiffServ framework are handling issues related to SLS and TCS.

Here, a description of the template Tequila suggests [many] and some examples of its usage done by Aquila [aquila] are presented. A similar template is elaborated on in AT&T's contribution [some].

6.2.1 TCS and SLS in a DiffServ Architecture

As the work in DiffServ WG progressed, it was agreed that the notions of SLAs and TCAs would be taken to represent the broader context, and that new terminology used to describe those elements of service and traffic conditioning is introduced – namely SLS and TCS.

The SLS may specify the packet classification and (re)-marking rules and also traffic profiles and actions to traffic streams that are within the traffic profile as well as traffic streams outside the traffic profile. The DS codepoint (DSCP) to be used for mapping into the various DiffServ classes may also be defined in the SLS. If multifield classifiers are used other elements in the packet header have to be used for classification. These elements have to be agreed on in the TCS.

A TCS has as its scope the acceptance and treatment of traffic meeting certain conditions and arriving from a peer domain on a certain link. More specifically, the TCS asserts that traffic of a given DiffServ class, meeting specific policing conditions, entering the domain on a given link, will be treated according to a particular (set of) Per Hop Behaviours (PHBs) and if the destination of the traffic is not in the receiving domain, then the traffic will be passed on to another domain (which is on the path toward the destination according to the current routing table state) with which a similar (compatible and comparable) TCS exists specifying an equivalent (set of) PHB(s).

The parameters[10)] included in the TCS/SLS may be:

[10)] *Due to the unidirectional nature of connections, the two directions of a flow across the boundary will need to be considered separately.*

- Detailed service performance parameters such as packet loss, packet delay and jitter. Expected throughput may also be relevant;

- Constraints on the ingress and egress points at which the service is provided. The ingress and egress points indicate the scope of the service;

- Traffic profiles that must be adhered to for the requested service to be provided, such as token bucket parameters;

- Disposition of traffic submitted in excess of the specified profile;

- Marking services provided;

- Shaping services provided.

In addition to these parameters, the SLS may specify more general service characteristics such as:

- Availability/Reliability, which may include behaviour in the event of failures resulting in rerouting of traffic;

- Encryption services;

- Routing constraints;

- Authentication mechanisms;

- Mechanisms for monitoring and auditing the service;

- Responsibilities such as location of the equipment and functionality, action if the contract is broken, support capabilities;

- Pricing and billing mechanisms.

The metrics to be used for validating that the delivery of the service is according to the parameters in the TCS are studied in the IETF IPPM WG, and can be found in [rfc2330].

Quantitative and Qualitative Services

IETF uses the expressions quantitative and qualitative related to the service delivery. Services can be categorised as qualitative or quantitative depending on the type of performance parameters and guarantees offered.

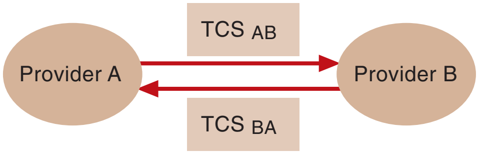


Figure 8 TCSs between two providers

Examples of qualitative services are:

1. Traffic offered at service level A will be delivered with low latency;

2. Traffic offered at service level B will be delivered with low loss.

This assurance is only relative and can only be verified by comparison.

Examples of quantitative services are:

1. 90 % of the traffic within the profile delivered at service level C will experience no more than 50 ms latency;

2. 95 % of the traffic within the profile delivered at service level D will be delivered.

In general, when a provider offers a quantitative service, it is necessary to specify quantitative policing profiles.

Quantitative provisioning is not a trivial task. With knowledge of the network routing topology and the TCSs at the boundaries, it is possible to compute the resources required at each interior node to carry the quantitative traffic offered at the edges. Based on the result of these computations, interior nodes must be configured with sufficient capacity to accommodate the quantitative traffic that will arrive at the node and in addition leave sufficient capacity remaining to accommodate some amount of qualitative traffic. In addition to installing and configuring the appropriate capacity at each interface, it may be desirable to configure the policers to assure that the resources actually consumed by the higher priority quantitative traffic do not exceed the expectations. Since the traffic receiving qualitative services cannot be assumed to follow specific routes with the same predictability as the traffic receiving quantitative services, the provisioning of qualitative traffic is more difficult and parameters must be estimated based on heuristics, experience and preferably on real-time measurements.

Inter-domain Considerations

The TCS has been described primarily in the context of a single domain providing services to a customer. In general, customers are end-users and/or hosts that reside in different networks. These networks are interconnected by multiple domains and require that the service spans these domains. Making an SLS, it is important to consider the interaction of services provided in the various networks involved, rather than the service provided by a single domain.

The service provider is expected to negotiate bilateral agreements at each boundary node at which it connects to another network. Figure 8 illustrates two providers – A and B, the services they offer to each other, and their relationship captured in TCSs.

The technical aspects of these agreements that relate to the delivery of differentiated services are captured in two TCSs – (1) for the services provided by Provider A to B, (2) for services provided by Provider B to A. Similar to the analogue discussion on SLAs and dependencies on different interfaces, TCSs needed by a provider at any boundary will be dictated by TCSs negotiated at the other boundaries. Provider A may serve a number of customers with services terminating at various boundary points in Provider B's network. The TCS between Provider A and Provider B must represent the aggregate requirements of the TCSs of all Provider A's customers.

In order to provide end-to-end services to its customers, Provider A must be able to assure the support for these services across multiple domains, which requires several issues to be solved:

- The service provided by a certain domain may not be compatible with the services provided by neighbouring domains;

- The services provided by a domain may be compatible with the services provided by neighbouring domains, but the PHB used to obtain the service might be different;

- The PHB might be the same, but the codepoint used to request the PHB might be different;

- The PHB and the codepoint are the same but differences in provisioning and charging models result in different service.

Determination of compatible services and negotiation of PHB codepoints for requesting the services is required. This process may be greatly simplified by the provision of a set of universal services using universally recognised codepoints.

The extension of quantitative services across multiple domains will require more uniformity in the nature of services provided. Qualitative services may on the other hand be extended end-to-end by a concatenation of service elements that may vary from domain to domain. For example, one domain may base a qualitative service on a Weighted Fair Queueing (WFQ) scheme with Random Early Discard (RED), while another may use priority queuing with RIO. Since the assurance end-to-end is looser it is possible that a meaningful service can be provided end-to-end by concatenating these two types of services.

A host may be directly attached to a differentiated service domain. Legacy hosts are unlikely to perform marking of their packets into Diff-Serv classes and are also unlikely to shape or police their traffic. These services may be provided on behalf of the customer. The policies used for marking and shaping have to be negotiated at the time the agreement is made between the network provider and the customer (host). Newer hosts may be capable of marking and traffic shaping. The overall resource constraints in the agreements may likely be somewhat static in this case. The host determines the manner in which the host shares these resources among its various traffic flows. The provider still has to configure policers to assure that the host does not seize more than its share of the resources or the amount of traffic in the various classes than agreed in the SLS or TCS.

6.2.2 Tequila – Traffic Engineering for Quality of Service in the Internet, at Large Scale

The aim of the Tequila draft [tequila] is to identify the basic information to be included in SLS considering the deployment of value-added IP service offerings over the Internet. When such IP service offerings are provided with a given QoS, the QoS should be defined in such an SLS from a technical perspective. Since these IP services are likely to be provided over the whole Internet, their corresponding QoS will be based upon a set of technical parameters that both customers and service providers will have to agree upon. Having this perspective, this draft aims at listing (and promoting a standard formalism for) a set of basic parameters that will actually compose the elementary contents of an SLS.

The Tequila specification effort tries to address the following:

- Provide a standard set of information to be negotiated between a customer and a service provider or amongst service providers within the context of processing an SLS;

- Provide the corresponding semantics of such information, so that it might be appropriately modelled and processed by the above mentioned parties (in an automated fashion).

It seems useful to consider the specification of an SLS template that these service providers would agree upon, so as to enforce an inter-domain QoS policy.

It is necessary to be able to allow for a highly developed level of automation and dynamic negotiation of SLSs between customers and providers. Automation and dynamics are helpful in providing customers (as well as providers) with the technical means for the dynamic provisioning of QoS. The work on SLS in Tequila is focused on an IP network that will be composed of DiffServ-aware network elements able to implement PHBs like Assured Forwarding (AF) and Expected Forwarding (EF).

SLS Template in Tequila

1. **Scope** – The scope indicates where the QoS policy is to be enforced. The geographical/topological region is indicated by the boundaries of the region. An SLS is associated with unidirectional traffic flows. A couple of ingress and egress interfaces denote the entry/exit points of the IP packets. The ingress and egress may be:
   - One-to-one;
   - One-to-many;
   - One-to-any;
   - Many-to-one;
   - Any-to-one.

   Many-to-many is excluded from the list but may be decomposed into many times one-to-many.

2. **Flow description** – Indicates for which IP packets the QoS guarantees are to be enforced. A flow description identifies a stream of IP datagrams sharing at least one common characteristic. An SLS contains one (and only one) flow description, which may formally be specified by providing one or more of the following attributes:

   - Differentiated Services information = DSCP;

   - Source information = source address;

   - Destination information = destination address;

   - Application information = protocol number, source port, destination port.

   The flow description provides the necessary information for classifying packets at a DS boundary node. BA classification requires only DSCP codepoint, while MF classification requires that the other information is given.

3. **Traffic envelope and traffic conformance** – Describes the traffic characteristics of the IP packet stream identified by the flow description.

   A binary traffic conformance identifies in-profile and out-of-profile (or excess) packets of an IP packet stream. Multi-level traffic conformance gives the opportunity for tagging the packets when reaching various threshold values.

   Traffic conformance parameters:
   - Peak rate (bits per second);
   - Token bucket rate (bits per second);
   - Bucket depth (bytes);
   - Maximum transport unit (bytes);
   - Minimum packet size (bytes).

4. **Excess treatment** – Excess treatment describes how the service provider will process excess traffic, i.e. out-of-profile traffic (in case of binary conformance testing) or n-level traffic (in case of n-level conformance testing).

   Excess traffic may be dropped, shaped and/or remarked.

   - *Dropped* – if excess traffic is dropped, then all packets marked as ‘out-of-profile’ by the Traffic Conformance Algorithm are dropped. No extra parameters are needed.

   - *Shaped* – if excess traffic is shaped, then all packets marked as ‘out-of-profile’ by the Traffic Conformance Algorithm are delayed until they are ‘in-profile’. The shaping rate is the policing/token bucket rate. The extra parameter is the buffer size of the shaper.

   - *Marked* – if excess traffic is marked or remarked, then all packets marked as ‘out-of-profile’ by the Traffic Conformance Algorithm are (re-)marked with a particular DSCP-value (yellow or red). The extra parameter is the DSCP.

   The SLS must specify the appropriate action, otherwise the excess traffic is dropped.

5. **Performance guarantees** – Performance guarantees describe the service guarantees offered to the packet stream identified by the flow description.

   There are four performance parameters:
   - delay, time interval, optional quantile;
   - jitter, time interval, optional quantile;
   - packet loss, time interval;
   - throughput, time interval.

   Some further details on these parameters are given in the following.

   Delay, jitter and packet loss guarantees are for the in-profile traffic in case of binary conformance testing. Having multi-level conformance testing, delay, jitter and loss guarantees may be specified for each conformance level respectively, except the last one. For example having three levels, one can have a delay guarantee for the “conformance level-1” packets and a different delay guarantee for the “conformance level-2” packets. No guarantees are given for excess ("conformance level-3") traffic.

The throughput is an overall guarantee for the IP packet stream, independent of a particular level.

The delay and jitter respectively indicate the maximum packet transfer delay and packet transfer delay variation from ingress to egress, measured over (any) time period with a length equal to the (indicated) time interval.

Delay and jitter may either be specified as the worst case (deterministic) bounds or as quantiles. Indeed, the worst-case delay/jitter bounds will be very rare events and customers may find measurements of e.g. 99.5th percentile a more relevant empirical gauge of delay/jitter.

The packet loss probability is the ratio of the lost (in-profile) packets between ingress and egress and the offered (in-profile) packets at ingress.

The ratio is measured over (any) time period with a length equal to the (indicated) time interval.

The throughput is the rate measured at egress counting all packets identified by the flow description. Notice that all packets, independent of their conformance level (in/out-of-profile) contribution. Indeed, if the customer (only) wants a throughput guarantee, then they do not care whether in- or out-of-profile packets are dropped, but are only interested in the overall throughput of the packet stream.

Quantitative performance guarantees – A performance parameter is said to be quantified if its value is specified to a numeric (quantitative) value. The service guarantee described by the SLS is said to be quantitative if at least one of the four performance parameters is quantified.

Qualitative performance guarantees – If none of the SLS performance parameters are quantified, then the performance parameters delay and packet loss may be qualified. Possible qualitative values (for delay and/or loss): high, medium, low.

6. **Service schedule** indicates the start time and end time of the service, i.e. when is the service available. This might be expressed as a collection of the following parameters:
   - Time of day range;
   - Day of the week range;
   - Month of the year range.

7. **Reliability** indicates the maximum allowed Mean Down Time (MDT) per year and the Maximum allowed Time To Repair (TTR) in case of service breakdown (e.g. in case of cable cut). The MDT might be expressed in minutes per year and the TTR might be expressed in seconds.

8. **Other parameters** such as route, reporting guarantees, security etc. are for further study by tequila.

SLS Negotiation Requirements

A major goal of the availability of an SLS template is to help in the deployment of dynamic SLS negotiation procedures between customers and providers or between providers. The Tequila draft mainly discusses the SLS template and its basic contents. The SLS negotiation protocol is for further study; however, a number of conditions that should be met by an SLS negotiation protocol are listed:

- Original service requests, according to the components of the specified SLS;

- Service acknowledgement (ACK), indicating agreement with the requested service level;

- Service rejection (NAC) but indicating the possibility of offering a closely related service (or indication of alternative DSCP to use for a particular service). The reply message may indicate the related offering by overwriting the proposed SLS attributes;

- Service rejection (REJECT) indicating incapability of providing the service;

- The ACK/NACK procedures require a reliable transport mode for such a negotiation protocol;

- Service modification from both user and provider.

6.2.3 Aquila – Adaptive Resource Control for QoS Using an IP-based Layered Architecture

The IST Aquila consortium [aquila] aims to have a standard formalised representation of SLS between the customer and the network. This representation should be very general and capable of expressing all the possible service offerings based on the DiffServ model. The Aquila consortium also identified the need for a mechanism to simplify the generic description of the SLS. This led to the definition of predefined SLS types.

The Aquila draft [aquila] is aligned with the Tequila-draft [many], and there is a set of commonalties between the Aquila and Tequila

Attribute	Measurement unit
Quantitative maximum Delay	(ms)
Quantitative maximum Jitter	(ms)
Quantitative maximum Loss Ratio	
Quantitative Delay percentile	(percentile; ms)
Quantitative Jitter percentile	(percentile; ms)
Quantitative mean Delay	(ms)
Quantitative mean Jitter	(ms)
Qualitative Delay	(medium/low/very low)
Qualitative Jitter	(medium/low/very low)
Qualitative Loss Ratio	(medium/low/very low)

Table 1 Performance guarantee attributes

approaches. The main difference is that the Aquila consortium has introduced the concept of predefined SLS types that are based on the generic SLS definition. These predefined SLS types can be used to simplify the interaction between the customer and the network. From the applications viewpoint, a predefined SLS type supports a range of applications that have similar communication behaviour and therefore similar QoS requirements, such as for delay, packet loss, etc. From an operator's point of view it simplifies the network management and allows efficient flow aggregation.

In a DiffServ network the SLS parameters should be used to map the user requirements into internal QoS mechanisms (e.g. DiffServ classes). The mapping process between the generic SLS and the concrete QoS mechanisms can be very complex if the user can freely select and combine the parameters. Naturally, in a DiffServ network there will be a restricted number of service classes handled in the core.

The SLS type in Aquila distinguishes between a customised SLS and a predefined SLS type. In the instance of a customised SLS all the parameters can be specified, whereas in the instance of a predefined SLS only a subset of the parameters must be specified. The predefined SLS types in Aquila are:

- PCBR – Premium CBR;
- PVBR – Premium VBR;
- PMM – Premium MultiMedia;
- PMC – Premium Mission Critical.

Both quantitative and qualitative performance guarantee attributes are foreseen. The quantitative values can be expressed as maximum values, mean values or percentiles. The qualitative attributes can be used to express relative guarantees between different classes.

The delay is meant as the one-way delay, the jitter is defined as the variation of one-way delay (max delay – min delay) of a flow. The details of the measurement procedure to evaluate statistics parameters like percentiles or mean values should be defined.

6.3 TeleManagement Forum (TMF)

As mentioned before, the content of an agreement may differ depending on the interface it relates to. In other words, an agreement between a user and a service provider (SP) would differ from the agreement between two service providers. When considering an SP-SP agreement, Telecom Management Forum (ex NMF) defined business models and related processes which could be used to define the potential content of the different SLA types.

The work done in NMF considers so-called end-to-end SLA[11], which should contain agreements about the following issues/topics:

- The service type and customised service template;
- Definition of common business processes (e.g. in the context of NMF Business Process model);
- Common QoS needs;
- Technical constraints;
- Definition of relevant QoS/performance parameters for the end-to-end relationship;
- Notifications and actions in case of problems;
- References to management interface types being supported (e.g. X.user, X.coop);
- Common management policies;
- Common security requirements, methods, policies;
- Common trouble administration interfaces, policies;

[11] *As mentioned before, there are numerous definitions of the agreement, SLA, etc. The one from NMF on the "end-to-end SLA" is "an SLA between multiple SPs, which defines common agreements between all parties involved in the service provisioning/consuming process" [NMF701].*

- Common accounting interfaces;
- Common interoperability tests and test suites if interworking between all SPs is necessary;
- Etc.

Relevant information related to the management of SLAs can be found in [GB917], where the issues of creating SLAs and examples of managing SLAs are handled.

6.4 EURESCOM

EURESCOM has a long tradition of running projects dealing with QoS aspects. Regarding SLA, a common QoS framework for dealing with QoS/NP in a multi-provider environment was developed. The terminology harmonising the understanding between different teams of experts included in the QoS-related work was settled. In addition, the concept of "one-stop responsibility" was introduced, and the applicability of the framework is exemplified for the VoIP service case. Some of the material included in this document is developed with those generic principles in mind. More details on their results can be found in [P806-site].

The P906-GI project [P906-site] handled the means of measuring and managing several classes offered to different application categories. The SLA is considered as a tool to resolve the responsibilities and achieve QoS end-to-end of the provider's network. The multi-provision is handled only related to the charging of the retail/wholesale services. The concept of Service Offer Specification (SOS) is introduced, where a user is offered:

- Network Parameters Level (NPL) reflecting delay, jitter, loss;
- NPL's guarantees probability;
- Traffic profile;
- Charge/price.

The values of the parameters and their significance are decided by the provider by relating application categories (AC) (e.g. interactive real-time applications) and quality categories (QC) (different classes possible to assure by the provider). More details on the QUASI-model, where the (QC, AC) mapping is given, and SOS presented in detail can be found in [P960d1], [p906ti6].

7 Relating SLAs and SLSs

As already presented, the service description within an SLA describes the service the customer may expect to have delivered from the provider. Each service should be reflected both in a related SLA and in SLS(s). Recall that each SLA includes the service description part and QoS-related part (in the following called SLS for short) as illustrated in Figure 9, where the technical aspects of the service provided are defined.

Mapping between SLA related to a service on the one hand and the corresponding SLS related to the QoS mechanisms on the other hand is not a simple one-to-one mapping. This is a challenge when designing SLAs since it involves the decision on the selection and implementation of QoS mechanisms to be used in the network, possible ways of relating and combining them to ensure the delivery of the service according to the agreement. In addition to the decision of QoS parameters, the QoS mechanisms need to be tuned properly so the resources are efficiently used during the service provision.

The challenge gets more complex where multiple providers are involved in the service provision. In that case, the primary provider (that is responsible for the service delivery towards the user) has to rely upon the service/QoS provided by a network they do not have control of.

Another challenge arises when a service provider provides multiple services using a single network, which is often the case with services offered in IP-based networks like VoIP and Video on Demand (VoD). The SLA and SLS may then be related in a number of ways, some of them identified and elaborated on in the following:

- *One-to-one:* Each service has a separate SLA with a description of QoS both at the application/service level and the network level, e.g. including both the required QoS parameter values at the application/service level and the required QoS parameter values at the IP level.
- *Many-to-one:* Each service has a separate SLA with a description of the QoS parameters at the application/service level. A set of services is provided and this set of SLAs share

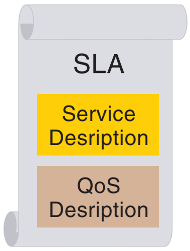


Figure 9 SLA contains both service description and QoS description

Object	Notation
Service A	The service name, as given and described in the service description included in the SLA
QoSa	QoS description related to service A at the application/service level (e.g. VoIP)
QoSIPa	QoS description related to service A at the common IP level (i.e. IP connectivity level)

Table 2 Notation used when analysing SLA/SLS mapping

one common QoS description for the service provided at the network level, i.e. the IP level. The services (e.g. VoIP and VoD) have distinct QoS descriptions at the application/service level (i.e. Voice and Video level). However, the services are reflected in a common QoS description at the network level, i.e. IP level.

- *All-in-one:* In this case the SLA relates to the bundled service, i.e. all services offered to the user and their quality are described in a single document. This might be the case if one provider is responsible for all services delivered and controls all networks involved in the service delivery.

In the following the examples of relationships between SLAs and SLS are elaborated on. The notation used is given in Table 2.

7.1 One-to-one Relationship between SLA and SLS

Assume three services – A, B, C, each of them having a separate SLA. Each of these SLAs includes QoS descriptions of all service components that are enveloped in an SLS. This SLS gives a full description of the service and relevant QoS parameters for all service components. In this case, SLS has to include both QoS parameters and values for the IP flows carrying the traffic generated by the application/service (e.g. IP flows for service A – QoSIPa) and the QoS parameters related to the application/service level, (e.g. for service A – QoSa). In this case the SLAs are separate from each other as illustrated in Figure 10.

In Figure 11 is illustrated an example of a service delivery with related agreements. The service provider (SP) delivers a VoIP service to the user. In this example, the SP is responsible for the IP connection used to deliver the VoIP service. The user and the SP have set up the SLA, which covers both business and technical aspects for the VoIP service delivery/usage. That means that the SP has to take care of the IP connectivity (via another SLA made with network operator, NO[12]), and to include the description and QoS-related aspects in the user-SP SLA. The QoS description in this SLA (QoSb) covers the relevant parameters for both IP network connectivity and VoIP level. In order to fulfil SLA User-SP, two other SLAs should exist, User NP and NP-SP. In this case a single provider (i.e. SP) controls the process of making agreements and has an overview/control of mapping service parameters and QoS parameters, but it is still not a trivial problem.

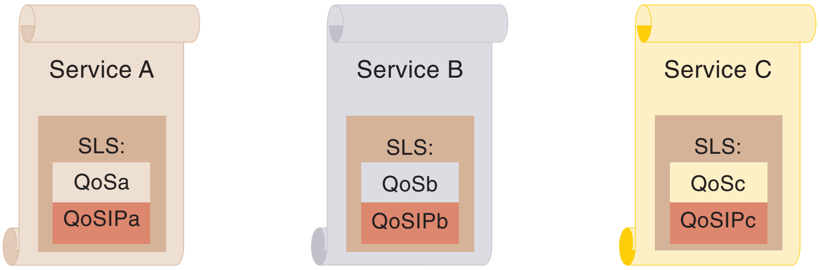


Figure 10 One-to-one relationship

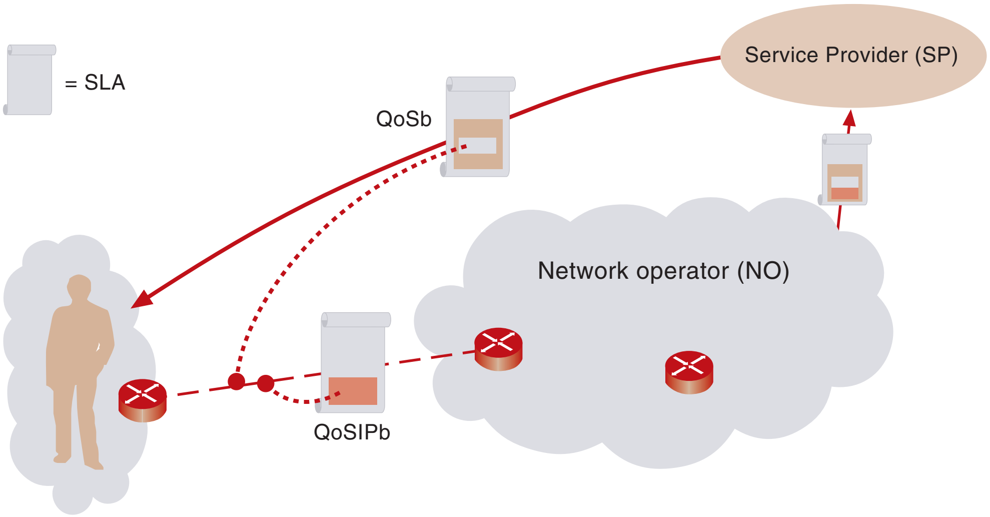


Figure 11 One service delivered and produced by a single provider

[12] *Note that the SLA made between SP and NO is not visible to the user and is therefore omitted in Figure 11.*

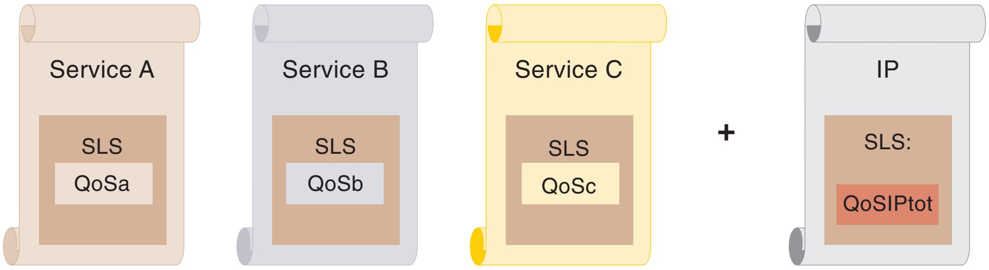


Figure 12 Many-to-one relation

7.2 Many-to-one Relationship between SLA and SLS

In a many-to-one relationship case the SLA gives the description of the service and all QoS parameters necessary to describe the service to be delivered at the service level, i.e. service A with QoS guarantee QoSa at the service level. The result is three SLSs for QoS description of application service levels (e.g. VoIP, Video). However, several services may be provided to a user using the same IP network and the same access. Therefore, an SLA may include a common SLS describing the common component, i.e. IP connectivity, and its QoS description is included in a separate SLS as illustrated in Figure 12 for the total set of services. In order to guarantee the services and the QoS described in separate SLAs the decision on the performance needed from a common IP service should consider these SLAs (service descriptions), QoS descriptions and traffic profiles. Combining these for the total set of services to be provided using the same IP network service, the resulting QoS parameters, QoSIPtot, may be found. QoSIPtot is related to the IP level and should be given in a separate SLS for the IP service.

Consider an example where SP B offers a service B (e.g. VoIP) to the user, but without taking care of the IP connectivity needed for the VoIP service delivery. The user and SP B agree the SLA, with the QoS description QoSb. At the same time user A agrees delivery of service A (e.g. VoD) with SP A as agreed in the SLA with QoSa. In addition, the user makes an SLA with the NO for the IP connectivity as illustrated in Figure 13. This service will be used for delivery of services A and B, i.e. typically there will be several services using the same resources towards the same user. Having a situation like this, it is very complex for the NO to monitor the service delivery and SLA assurance, and to report to each of the SPs the performance of their particular services. The problems are partially solved if the SPs make SLAs with NO for IP services delivered to the user (thus hiding NO from the user, and including the relevant information in the user's SLA), but even then the situation is not simple.

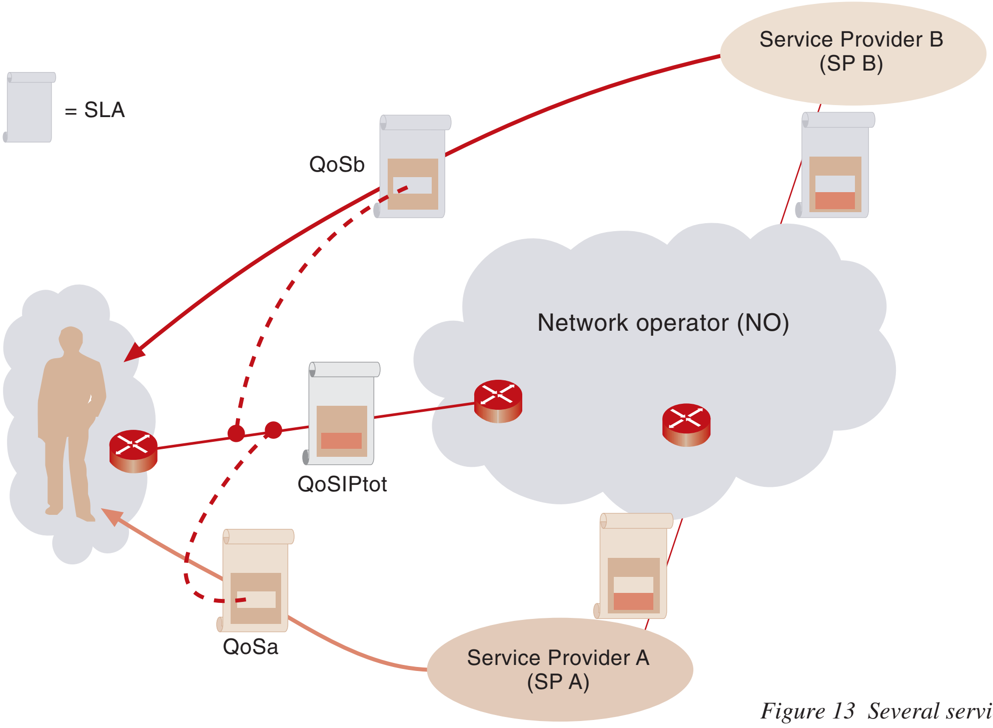


Figure 13 Several services delivered over the same access

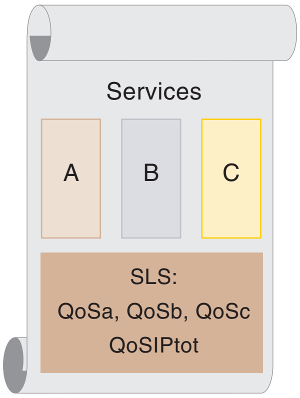


Figure 14 All-in-one relationship

7.3 All-in-one Relationship between SLA and SLS

In this case, the SLA includes all IP-based services provided using the same IP service. A full description of the QoS parameters that are necessary on the application/service level for all the IP-based services is given. Bearing in mind their QoS parameters and traffic profile, the QoS parameters and values of the IP basic service may be described as one QoS description or several (depending on the way the basic service is used).

As illustrated in Figure 14 a single SLA envelops the service descriptions for all services (A, B, C), as well as one SLS which includes QoS descriptions for each of the services (i.e. QoSa, QoSb and QoSc), as well as a QoS description of the common IP service QoSIPtot. Note that QoSIPtot includes the requirements and parameters from all QoS descriptions and demands given in QoSa, QoSb and QoSc.

In the case of a provider offering bundled services, i.e. usually combining several services into one common (bundled) service delivery, this type of SLA is very important. All-in-one relationships are traditionally present in the case of monopolistic operators that are responsible for the full service package, i.e. all services delivered to the users. The user would then have an agreement with a single (so-called primary) provider that will be responsible for the delivery of other services offered by other providers (e.g. service provider, content provider, etc.) that are using the connectivity service offered by the primary provider.

Figure 15 illustrates the case where an NO is the primary provider for the user and he needs to take care and agree the SLAs with all the providers (e.g. SP A, SP B) that are contributing to the service(s) delivered to their users. In such a case the "all-in-one" relationship is usually present.

8 Concluding Remarks

While numerous research projects are trying to solve the support of the future QoS-aware IP services by introducing the concept of SLAs, the practice is rather unclear.

Existing SLAs involve no 'hard' QoS guarantees for IP-based services, where the parameters are strictly defined, objectively measurable with the desired granularity and where the values are set to the theoretical estimations. The reason is obviously to be found in the fact that the technology is not mature enough for it to be fully

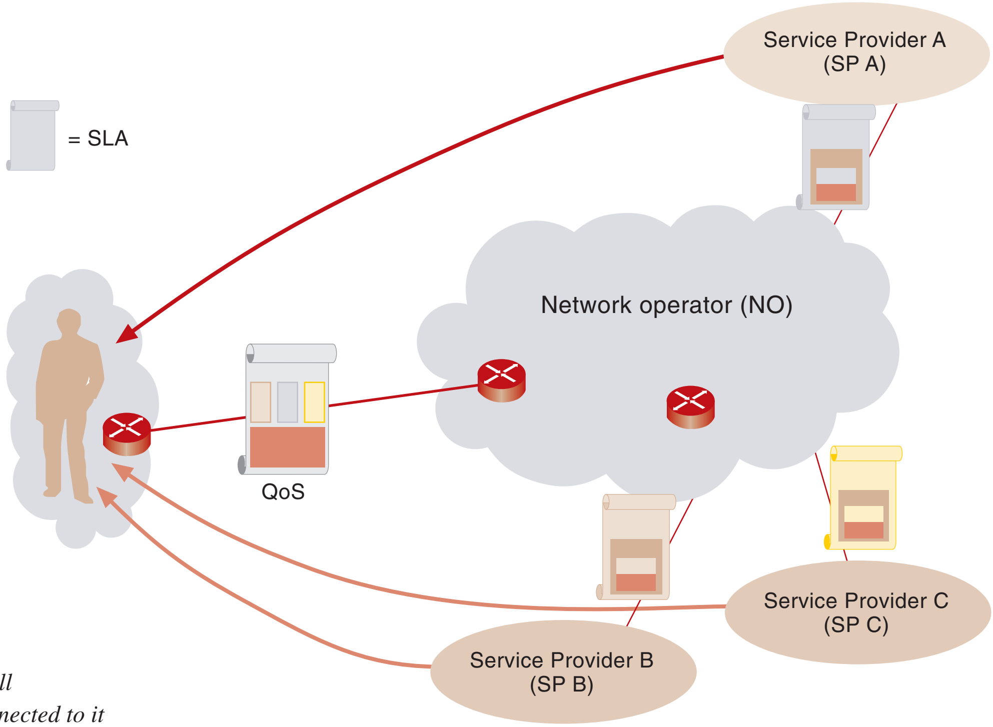


Figure 15 NO is responsible for all services delivered to the users connected to it

supported. Research and the trials of different implementations are improving the knowledge, but the real-life implementation causes the 'softness' of SLAs. In short, today's SLAs cannot promise more than what is technically possible to support. However, this fact does not mean that the future is not very bright for the SLA. When the technology is evolving the SLAs have a possibility of including even 'hard' QoS guarantees. Acquiring more experience and deciding on the mechanisms that are crucial and optimal, not only differentiated services may be offered, but also the QoS guarantees can be assured. This implies that the content of SLAs would include sharpened values.

As shown in the examples of existing SLAs presented in Chapter 4, the selection of QoS parameters describes the performance of communications between any two points within a network. The way of realising and handling SLAs are typically of a cloud type since that is the easiest to specify and support. After introducing the additional features/functionality in the IP-based network, moving towards stricter QoS guarantees will be feasible and the SLAs would be of the tunnel/funnel types. The granularity could cover a single packet, flow, session, monthly subscription, 10-year agreement, etc.

Naturally, no prediction can be certain and obviously many issues are still open both related to the usage and applicability of the mechanisms to support the QoS architecture [rfc2990]. But it is given in different surveys that it is better to give some guarantees and feedback to the customers rather than to lean on best effort services for a provider who wants to build/keep brand-name and attract/preserve loyalty of customers. Therefore, SLAs undoubtedly have a future, both in single- and multi-provider situations, although there are still many open issues to be studied further. In the case where a single provider provides its services by crossing only one administrative domain that he owns/controls, the SLA is still needed to formalise the behaviour of customers and the resulting expectations related to QoS.

References

[aquila] Salsano, S et al. 2000. *Definition and usage of SLSs in the AQUILA consortium.* Internet Draft. draft-salsano-aquila-sls-00.txt.

[Cain97] Caine, A. *Negotiating an Effective Service Level Agreement.* Gilbert&Tobin, 1997. http://www.gtlaw.com.au

[diffserv] *IETF DiffServ WG charter.* (2001, October 8) [online] – URL: http://www.ietf.org/html.charters/diffserv-charter.html

[E.801] ITU-T. *Framework for service quality agreement.* Geneva, 1996. (ITU-T E.801.)

[epoch] *Epoch Internet web site.* (2001, October 8) [online] – URL: http://www.epoch.net

[epoch-a] *Epoch Internet SLA-related web site.* (2001, October 8) [online] – URL: http://www.epoch.net/corpinfo/sla_access.html

[epoch-sla] Epoch Internet SLA-related web site. URL: http://www.epoch.net/corpinfo/agreements.html

[epoch-w] Epoch Internet SLA-related web site. URL: http://www.epoch.net/corpinfo/sla_hosting_colo.html

[ETR003] ETSI. *Network Aspects: general aspects of Quality of Service and Network Performance.* Sophia Antipolis, 1994. (ETR 003.) (ref. RTR/NA-042102).

[GB917] TMF. *GB917 SLA Management Handbook, Member Evaluation version 1.0.* November 2000. (TMF GB917.)

[Gray00] Gray, J. *Negotiating An Effective Service Level Agreement – II.* Sydney, Gilbert& Tobin, 2000.

[I2-site] Internet 2 project site. URL: http://www.internet2.edu/html/about.html

[I.350] ITU-T. *General Aspects of Quality of Service and Network Performance in Digital Networks, including ISDN.* Geneva, 1993. (ITU-T I.350.)

[id-term] Grosmann, D. *New terminology for diffserv.* draft-ietf-diffserv-new-terms-04.txt. IETF, March 2001.

[many] IETF. T'Jones, Y et al. *Service Level Specification and Usage Framework.* Internet Draft, October 2000. http://www.ietf.org/internet-drafts/draft-manyfolks-sls-framework-00.txt

[NMF701] Network Management Forum. *Performance Reporting Definitions.* 1997. (NMF 701.)

[policy] *IETF policy WG charter.* http://www.ietf.org/html.charters/policy-charter.html

[P806d1] EURESCOM. *EQoS – A common framework for QoS/NP in a multi-provider environment.* Heidelberg, 1999. (EURESCOM P806-GI Deliverable 1.)

[P806-site] *EURESCOM P806-GI site.* [online] – URL: http://www.eurescom.de/public/projects/projectstartframe.asp

[P906-site] *EURESCOM P906-GI site.* [online] – URL: http://www.eurescom.de/public/projects/projectstartframe.asp

[P906d1] EURESCOM. *P906-GI QUASIMODO Deliverable 1: Offering QoS classes to end-users.* May 2000. http://www.eurescom.de/public/projectresults/P900-series/906d1.htm

[p906ti6] Bostica, P (ed.). *Summary of QUASIMODO findings on QoS.* [online] – URL: http://148.121.27.136/public/projectresults/P900-series/P906ti6.htm

[rfc1633] IETF. Braden, R et al. *Integrated Services in the Internet Architecture: an Overview.* 1994. (RFC1633.)

[rfc2123] IETF. Brownlee, N. *Traffic Flow Measurement. Experiences with NeTraMet.* 1997. (RFC 2123.)

[rfc2330] IETF. Paxon, V et al. *Framework for IP Performance Metrics.* 1998. (RFC 2330.)

[rfc2474] IETF. Nichols, K et al. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.* 1998. (RFC 2474.)

[rfc2475] IETF. Blake, S et al. *An Architecture for Differentiated Services.* 1998. (RFC 2475.)

[rfc2543] Handley, M et al. *SIP: Session Initiation Protocol.* 1999. (RFC 2543.)

[rfc2597] IETF. Heinanen, J. *Assured Forwarding PHB Group.* 1999. (RFC 2597.)

[rfc2990] IETF. Huston, G. *Next Steps for the IP QoS Architecture.* 2000. (RFC 2990.)

[sip] Johnston, A B. *SIP – Understanding Session Initiation Protocol.* Artech House, 2001.

[some] Rajan, R, Celenti, E, Dutta, S. *Service Level Specification for Inter-domain QoS negotiation.* 2000. [online] – URL: http://www.ietf.org/internet-drafts/draft-somefolks-sls-00.txt

[Tele0200] Espvik, O et al. Managing QoS in Multi-Provider Environment – a Framework and Further Challenges. *Telektronikk*, 96 (2), 71–79, 2000.

[tequila] IETF. Goderis, D et al. *Service Level Specification Semantics, Parameters and negotiation requirements.* Internet Draft, 2000. draft-tequila-diffserv-sls-00.txt

[top100] UUNet site – Recognition article: http://www.uk.uu.net/company/recognition/index.asp

[uunet] *UUNet SLA.* (2001, October 8) [online] – URL: http://www.uu.net/customers/sla/terms

[uunet1] UUNet's Dial Up Access – Corporate level service. URL: http://www.uu.net/us/support/sla/servicessupported/dan.xml

[Verm99] Verma, D. *Supporting Service Level Agreements on IP Networks.* Indianapolis, Macmillan Tech., 1999.

[Y.1540] ITU-T. *Internet Protocol Data Communication Service – IP Packet Transfer and Availability Performance Parameters.* 1999. (ITU-T Y.1540.) (ex. I.380).

Modelling the Topology of IP Networks

TERJE HENRIKSEN, ANNE-GRETHE KÅRÅSEN AND STÅLE WOLLAND

Terje Henriksen (59) is Research Scientist at Telenor R&D. He has been involved in standardization and R&D projects within the area of network level modelling since 1991; from 1996 to 2001 as the rapporteur for Question 18 of SG4 within the ITU-T. He is working in the Internet Network Architecture group.

terje-fredrik.henriksen @telenor.com

Anne-Grethe Kåråsen (42) is Research Scientist, R&D Kjeller. She is working in the Internet Network Architecture group with special interest in layer 1-3 network management and control.

anne-grethe.karasen @telenor.com

1 Introduction

Topological information and the dissemination of such information are important issues when discussing the architecture of IP networks capable of delivering differentiated QoS. The purpose of this paper is to describe a network level model representing the topological aspects of such networks. It is based on efforts carried out within the TrafHan project.

Section 2 provides a description of networks exhibiting a varying degree of QoS support, ranging from no support (Best Effort) via support per traffic class (DiffServ or DiffServ-over-MPLS) to support per traffic flow (IntServ). Four major limitations related to the Best Effort service are identified and used as the basis for the assessment of the others. The current model describing topology in terms of subnetworks, topological links and connection point groups is introduced and so is the Traffic Trunk, an abstract representation of traffic in DiffServ-over-MPLS networks.

The modelling methodology utilized is found in Section 3. The description of the network resources build on the entities of the generic network architecture defined in [G.805]. The methodology is an enhanced version of the Reference Model for Open Distributed Processing [RM-ODP], adapted to fit with the modelling requirements for telecom networks.

The Enterprise Viewpoint, constituting the requirements part of the resulting model, is described in Section 4 in terms of common community policies and actions related to the resource types involved.

In Section 5 the deletion of a topological link is presented as an example in terms of Enterprise (requirements), Information (static behaviour), and Computational (dynamic behaviour) Viewpoint descriptions. This constitutes the communication protocol neutral part of the model. The protocol specific part belongs to the implementation and is not addressed. It would have been documented in the Engineering Viewpoint based on the elements of the specific communication protocol chosen.

2 Network Characteristics

In this section, we will describe the network topology being the basis for the formal model. Best Effort networks as well as networks offering services with quantifiable QoS are discussed. Rather than providing a full description, the emphasis has been on describing properties that are candidates for being visible as part of the model.

The functionality described in this section is more extensive than that exposed by the current model. This is because the level of detail of the final model is to be decided later in the TrafHan project.

2.1 Best Effort IP Networks

IP networks are located at the network layer in the OSI model. The core element of an IP network is the router. It consists of two main functions, the routing function and the forwarding function, confer Figure 1.

The forwarding function transfers packets between the input- and output ports on the basis of the information in the routing table. The entries of the routing table, also known as next hop information, are calculated by the routing function taking into account the destination address and the routing algorithm chosen. It does so by comparing the appropriate part of the des-

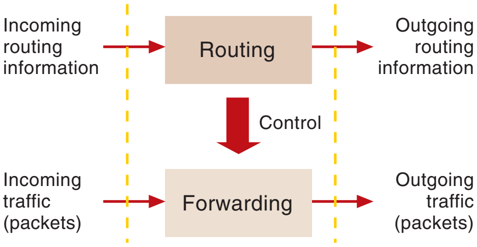

Incoming routing information
Routing
Outgoing routing information
Control
Incoming traffic (packets)
Forwarding
Outgoing traffic (packets)

Figure 1 The Best Effort IP router

tination address with the corresponding part of the address (longest prefix match) of every neighbouring router.

When a new router is installed and the source and destination of its in- and outgoing links are defined, this information is passed to all other routers by means of flooding. It is stored in the topology database of the router and used to recalculate the routing table entries affected by the introduction of the new router. In this way, each router holds information about the existence and topology of every other reachable router, i.e. it has a view of the overall network.

Within the domain controlled by a single ISP, i.e. within an Autonomous System, an Interior Gateway Protocol like the Open Shortest Path First – or the Intermediate System-Intermediate System protocol is utilized. For interdomain routing, a Border Gateway Protocol (BGP), such as BGP-4 is used.

The forwarding function is realized as a FIFO queue with tail-first dropping, i.e. in case of congestion, the last incoming packet is dropped first. There is only one queue for each output port and no marking of packets, so every packet gets the same treatment. Only one network service is provided, Best Effort, i.e. "as soon as possible" with "as much bandwidth as possible". There is no way of preventing greedy users ignoring congestion signals and making things worse for other users.

Most IP networks are Best Effort-based, despite a number of shortcomings encountered with such networks, in particular:

1. There is no way of enforcing differentiated behaviour to individual packets, flows or traffic aggregates.

2. There is no way of ensuring "fair" service provisioning, i.e. binding the effect of over-usage to the over-user.

3. There is no way of providing differentiated services with quantifiable QoS.

4. There is no way of utilizing other routes than the one chosen by the routing protocol, nor is it possible to provide load-sharing with slightly less favourable routes.

2.2 Traffic Engineering (TE) in IP Networks

Traffic Engineering, in this context, relates to the tools available to overcome the shortcomings of Best Effort networks as previously described. There are basically two options available for achieving that, in terms of two network service architectures:

- Integrated Services (IntServ);

- Differentiated Services (DiffServ), possible in combination with MultiProtocol Label Switching (MPLS) , i.e. DiffServ-over-MPLS.

Over-provisioning of network capacity may be utilized temporarily to cope with expected traffic increase, but it cannot solve any of the four shortcomings of the Best Effort service on a permanent basis.

An extensive discussion of the properties of IntServ, DiffServ and MPLS is given in [RessHandlIP].

2.2.1 Differentiated Services (DiffServ)

In DiffServ, aggregated packet flows are classified into Behaviour Aggregate (BA) traffic classes. The treatment to be applied to all the flows of a BA in every DiffServ router is called the Per Hop Behaviour (PHB). The classification may either be based on BA membership only, or on Multiple Fields (MF) membership. With the latter, a number of fields in the IP header, such as destination host address and port number, are used as additional basis for the classification. The BAs are distinguished from one another by the DiffServ Code Point (DSCP) values specified in the DS-byte of the IP4 header, replacing the TOS byte when DiffServ is being used.

When multiple BAs share an ordering constraint, i.e. the packets may not be re-ordered, these BAs form an Ordered Aggregate (OA). The corresponding set of PHBs is called a PHB Scheduling Class (PSC). Behaviour may also be defined on the AS level, so-called Per Domain Behaviour.

DiffServ presumes the existence of proper SLAs/SLSs to define the service level to be delivered to the users. The BA-/MF-classification is part of the SLS.

A number of PHBs is being defined by the IETF; Default Class (DC), Class Selector (CS), Expedited Forwarding (EF), Assured Forwarding1-4 (AF1-4).

EF provides forwarding characterized by low loss, low delay, low jitter and assured bandwidth.

AF aims at delivering packets within the agreed customer profile with high probability, whereas out-of-profile packets are delivered whenever available bandwidth permits. For each of the four AF classes, three levels of drop precedence

Ståle Wolland (55) is Research Fellow at Telenor R&D. He holds a PhD in Physics from Heriot-Watt University in 1974. He joined Telenor R&D in 1976 and has worked with information systems, distributed systems, network management systems, middleware and peer-to-peer systems.

stale.wolland@telenor.com

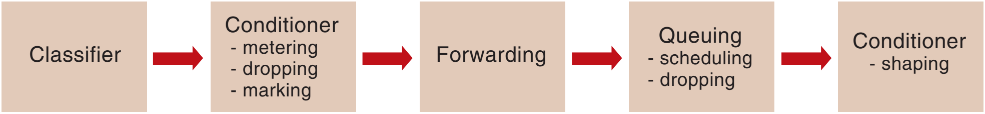


Figure 2 The DiffServ router

are defined. Each precedence level represents a separate PHB, and thus a reserved DSCP value.

The functional blocks of a DiffServ router are shown in Figure 2.

The functionality of the first two blocks is performed at the ingress of the DiffServ domain to enforce the SLA in question. MF-classification typically takes place at the edge of the network, whereas BA classification may take place in every DiffServ router.

2.2.2 Multi-Protocol Label Switching (MPLS) Combined with DiffServ

The router in an MPLS domain, the Label Switched Router (LSR), works on the basis of the MPLS header being attached to the packets at the ingress LSR, confer Figure 3. The Forwarding Equivalent Class (FEC) of the packets is decided on the basis of the destination address and the DiffServ PHB or PSC classification. The packets are encapsulated into an MPLS frame (with MPLS overhead added) and forwarded to the next LSR.

The route to be followed by the packets belonging to a particular traffic class (a traffic trunk) or set of traffic classes, i.e. the Label Switched Path (LSP), has been set up in advance by ordinary routing or constraint-based routing. The routing process is either controlled by RSVP or LDP (Label Distributed Path) signalling directly, or by invoking the same functionality at a management interface in the ingress LSR.

Each LSR contains a Label Information Base (LIB), supporting look-up of the outgoing interface as a function of the destination address.

Operations are available to merge traffic in the transit LSRs along the route.

At the egress LSR, the MPLS header is stripped off and further transport is again carried out on the basis of the information in the IP header.

MPLS in isolation does not help much in providing differentiated behaviour, fair service and quantifiable QoS. It is the combination with DiffServ that provides the requested differentiation of behaviour to traffic trunks being the basis for quantifiable performance guarantees with the service fairness preserved, and all this in an environment characterized by powerful and flexible means for routing traffic.

2.2.3 Integrated Services (IntServ)

An IntServ network provides specific classes of service to individual flows or groups of flows. In addition to Best Effort, two other classes are available:

- Controlled Load (CL). This class delivers low average delay and minimum loss. It resembles Best Effort service as if the network were unloaded.
- Guaranteed Service (GS). This service offers quantifiable bounded queuing delay and no loss. It is intended for real time applications with strict timing requirements.

In order to establish the flow-specific state, IntServ makes resource allocations in the routers by means of the Resource Reservation Protocol (RSVP). The PATH-message is transmitted downstream from the source host to the destination carrying with it the TSpec data structure. TSpec contains the requested delay, jitter and bandwidth parameters including the Token Bucket parameters. The path state established in each router includes the address of the upstream router. This datum is essential for the upstream transmission of the RESV-message. When the

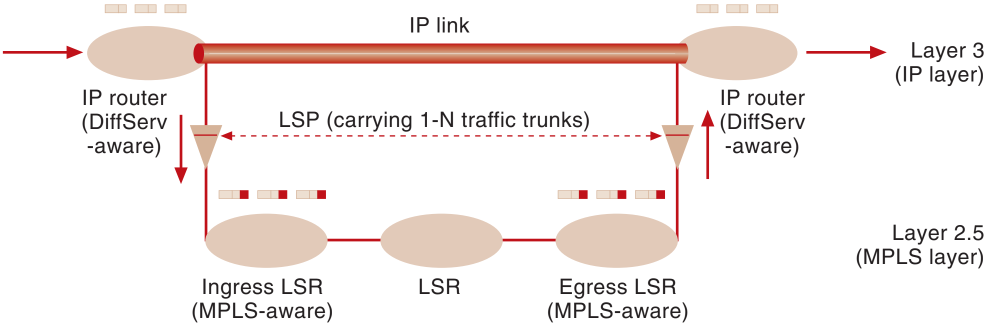


Figure 3 IP over MPLS

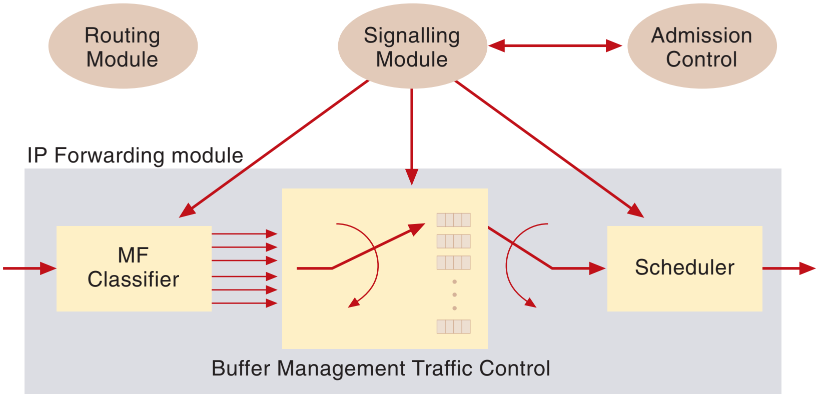


Figure 4 The IntServ router

PATH-message is received by the destination host, the RESV-message is created and transmitted upstream towards the source host, making the requested resource allocations in each router along the path, if possible. The receipt of the RESV-message in the source host signifies that consistent resource allocations are being made in every router, so that transmission of user data may begin.

The main functional blocks of an IntServ router are shown in Figure 4. When compared with the Best Effort router, the forwarding module has become more complex. It constitutes an MF (Multi-Field) classifier, mapping the incoming packets to a service class based on multiple fields in the packet header. The scheduler is managing the forwarding of flows from the output queues of the Buffer Management block.

In addition to the modules for routing and forwarding, there is a Signalling module dealing with the RSVP signalling messages, and an Admission Control module handling the resource allocation requests.

By basing the service provisioning on resource allocation and policing (in the scheduler) of the individual flows, IntServ is capable of providing differentiated service classes in a "fair" manner and with a quantifiable QoS. It overcomes the first three limitations of the Best Effort service. The main problem with IntServ is the extensive processing taking place in each router due to the per-flow handling of the resource allocation. For the same reason, the establishment of new paths to avoid bottlenecks in the network or support load-sharing or protection switching is less attractive, albeit possible.

2.3 Modelling the Network Topology

In the current topology model [GenIPmodel], the router is represented on the network level by a subnetwork with one connection point group (CPG) constituting a number of equal connection points (CPs) for each incoming and outgoing link[1] as shown in Figure 5. Each CPG corresponds to a set of physical ports.

When compared to the best-effort router in Figure 1, the function modelled is the routing. Forwarding is not part of the topological representation.

The interconnections between the routers are modelled as links with a certain capacity in terms of bandwidth. More than one interconnection may be transported over a single link.

The topological structure of an IP network may be modelled on the basis of subnetworks and links as shown in Figure 6. A Topological Link is a link exhibiting the additional constraint of being supported by a single trail in the server layer. For example, the IP link in Figure 4 is a topological link if it is supported by a single LSP in the MPLS layer.

Subnetworks are either interconnected locally by means of common CPGs or remotely via links.

2.3.1 Partitioning of Subnetworks

A more abstract representation of the network in Figure 6 is obtained by hiding the internal structure represented by the contained links and subnetworks, leaving the outer subnetwork and the CPGs at the boundary of the outer subnetwork only to be visible. This process may be recursively repeated, creating a more abstract representation each time. The inverse process, decomposing subnetworks into contained subnetwork groupings is called subnetwork partitioning.

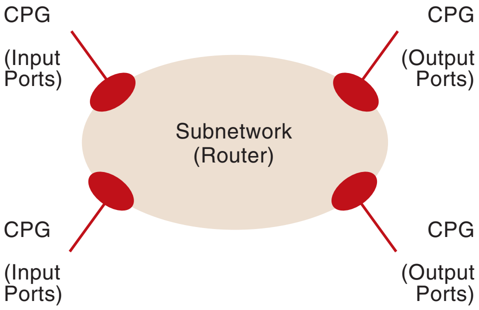


Figure 5 Router, network view

[1] *In many cases, all the input ports belonging to one incoming link are interconnected so there is only one input CP.*

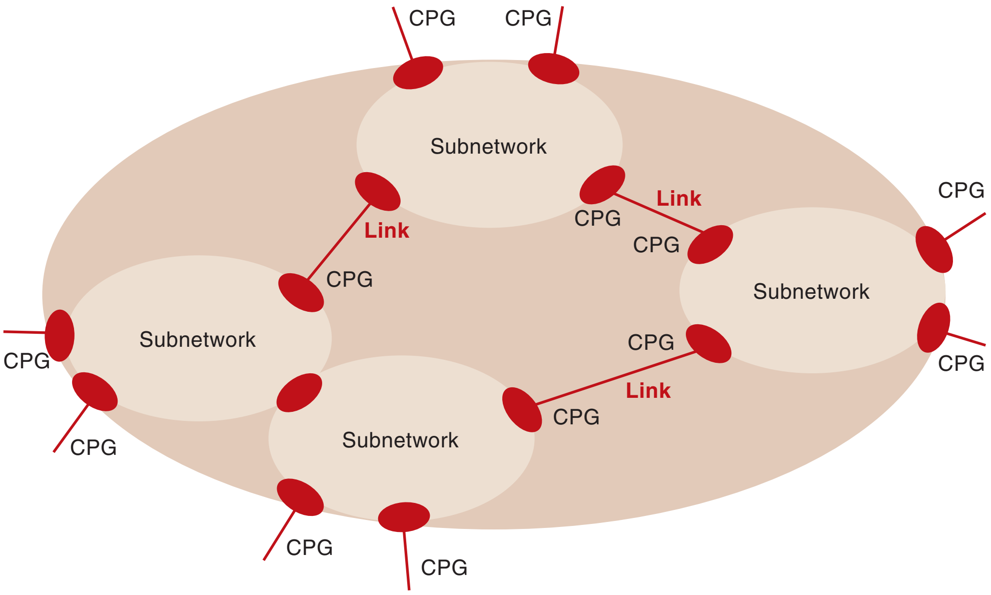


Figure 6 Network topology

To support partitioning, the model must define the mappings between the different levels of partitioning.

An important property of contained subnetworks is that they may span multiple physical locations.

Routing tables may be defined for containing subnetworks the same way as was done for the subnetwork representing a single router.

2.4 The Traffic Trunk

Traffic in the context dealt with here is either planned traffic for which network capacity must be calculated, measured traffic that needs analysis to explain the observed behaviour, or real traffic in an operational network.

Traffic is carried by unidirectional packets, each comprised by an IP header and an amount of data. A flow is a unidirectional stream of packets, distributed over time. It may have a very fine granularity reflecting a single interchange between hosts. These single interchanges are often called micro-flows. The aggregated flow is a number of flows that share forwarding state and a consistent resource reservation along a sequence of routers. The route of a flow across a subnetwork is an ordered list of CPGs at the next lower level of partitioning.

The traffic trunk is an abstract description of packet traffic. According to the definition in [PASTE], it constitutes a number of flows aggregated according to their traffic class (DiffServ-class) and placed inside an LSP together with zero (L-LSPs) or more (E-LSPs) other traffic trunks of different class.

The overall traffic requirements for the network in question is represented jointly in [TE_REQ] by the combination of the properties of the traffic trunks, topological constraints and the capabilities of the other routing protocols involved on layer 3 (IGP-oriented).

2.4.1 The Traffic Trunk Model

The properties of the traffic trunk may be expressed in terms of an object containing operations and attributes as described below.

```
TrafficTrunk “myGoodFriend”{
    OPERATIONS {
        Establish( );
            “An instance of a traffic trunk is
            created using this operation.”
        Activate( );
            “To cause the traffic trunk carrying
            traffic.”
        Deactivate( );
            “To stop a trunk carrying traffic.”
        Modify( );
            “To allow traffic trunk attributes
            being modified.”
        Reroute( );
            “To reroute a traffic trunk.”
        Destroy( );
            “To delete a traffic trunk including
            the release of all resources allocated
            to it, such as label space and
            available bandwidth.”
            };
    ATTRIBUTES {
        TrafficClass;
            “This is the traffic class according to
            the DiffServ classification.”
        DSCP;
            “This is the DiffServ Code Point of
            the traffic class.”
PHB_Definition;
    "This is the Per Hop Behaviour of
    the Behaviour Aggregate selected by
    the DSCP value. It is a complex data
    structure and it may possibly be
    represented as a separate object."
FEC;
    "The value of this attribute is given
    by the mapping from the traffic class
    to the forwarding behaviour given to
    the packets of this class in the MPLS
    domain."
TrafficParameters;
    "The traffic parameters specify the
    properties of the traffic trunk in
    terms of static and/or dynamic
    (traffic shaping) bitrates. As such,
    they reflect the resource
    requirements of the traffic trunk."
GenericPathSelectionAndMaintenance;
    ExplicitlyRouted_LSP_Creation;
        "The value of this attribute
        evaluates to "true" when ER_
        LSPs are required. When hop-
        by-hop setup governed by the
        underlying routing protocol is
        required the value evaluates to
        "false"."
    PathPreferenceRule;
        "This attribute has the value
        "Mandatory" when no other path
        may be used and "Optional"
        when alternative paths may be
        used. Path preference rules may
        be applied recursively upon a
        hierarchy of paths to cater for
        cases like rerouting when alarms
        occur."
    ResourceClassAffinity;
        "This attribute defines a
        sequence of resource class/
        affinity tuples where the affinity
        is the "applicable/not applicable"
        property for each resource class.
        Related to resource reservation
        during ER-LSP setup, this
        construct may e.g. support
        explicit inclusion/exclusion of
        certain links."
    Adaptivity;
        "The adaptivity attribute
        specifies whether re-optimisation
        of the path is permitted or not."
    LoadShare;
        "The amount of the overall
        traffic trunk carried by this
        sub-stream"
Priority;
    "The relative importance of the
    traffic trunk in question."
Preemption;
    "The preemption attribute specifies
    whether this traffic trunk can
    preempt lower priority traffic or not
    and whether this traffic trunk can be
    preempted by higher priority traffic
    or not."
BasicResilienceDecision;
    "The basic resilience decision
    attribute determines whether the
    traffic trunk is to be rerouted when
    a failure occurs."
ExtendedResilienceBehaviour;
    "The extended resilience behaviour
    attribute specifies additional
    behaviour to take place during a
    failure situation such as the choice
    of alternate paths to be used, if any."
Policing;
    "The policing attribute specifies the
    actions to be taken when non-
    compliant traffic parameters are
    provided by the traffic trunk, i.e.
    should it be rate-limited, tagged
    or forwarded without any action."
};
};
```

3 Modelling Methodology

The methodology chosen has been developed by the group of experts dealing with network level modelling within ITU, i.e. Question 18 of SG4. An overview can be found in [ITUmeth].

A generic functional architecture for transport networks as documented in [G.805] has been developed by SG13. This has been used in specialisations for SDH, ATM, WDM, Access Networks and (under development) connectionless communication (like IP).

The concepts described in [G.805] are essential to the understanding of the modelling methodology and they will be described in the following sub-section.

3.1 The Generic Network Architecture

A suitable abstraction level for dealing with topology management issues is the network level as opposed to the network element level. ITU has developed a generic network level transport functional model in the Rec. G.85x series that serves as a suitable starting point for a topology management model.

[G.805] provides a high level view of the network functions based on a small set of architectural entities (functional blocks) interconnected via reference points. Two main network representations may be provided on the basis of this architecture:

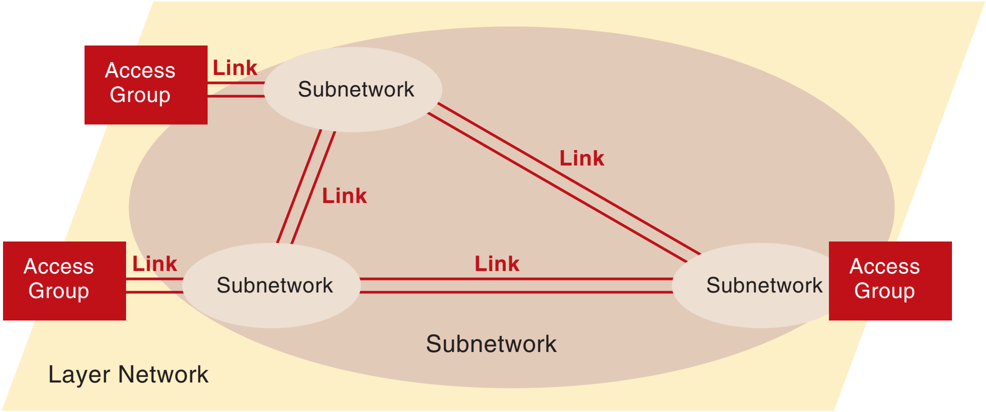


Figure 7 Network Architecture – the topological view

- Topology[2] in terms of links, subnetworks and access groups[3].

- Connectivity in terms of trails, link connections, subnetwork connections, ports and reference points.

Here, the topological representation will be addressed.

[G.805] introduces a set of concepts for describing the various functional aspects of a transport network:

Architectural component: An item that is used to generically describe transport network functionality.

Network: All of the entities (such as equipment, plant, facilities) which together provide communication services.

Transport network: The functional resources of the network which convey user information between locations.

Topological component: An architectural component, used to describe the transport network in terms of the topological relationships between sets of points within the same layer network. Here we focus on four topological components:

- Layer network;
- Subnetwork;
- Link;
- Access group.

Layer network: A "topological component" that includes both transport entities and transport processing functions that describe the generation, transport and termination of a particular characteristic information. A layer network is defined by the complete set of access groups of the same type which may be associated for the purpose of transferring information.

Subnetwork: A "topological component" used to effect routing of a specific characteristic information. A subnetwork is defined by the set of ports (see [G.805]) which are available for the purpose of transferring characteristic information. A subnetwork exists within a single layer network.

Link: A "topological component" which describes a fixed relationship between a "subnetwork" or "access group" and another "subnetwork" or "access group".

Access group: A group of co-located trail termination functions that are connected to the same subnetwork or link.

The topological view describes the geographical distribution of the resources of a single layer network. Layering is a method for splitting the overall transport function into a hierarchy of layer networks on top of each other where each layer uses the service from the server layer to provide its own service.

The topological concepts and their relationships are illustrated in Figure 7.

3.2 The Enhanced RM-ODP Framework

ITU-T SG4 has chosen the ISO Reference Model – Open Distributed Processing (RM-ODP) [X.901, X.902, X.903, X.904] as the basic framework for management modelling on the

[2] *Of a layer network.*

[3] *When modelling the topology of a subnetwork rather than a full layer network, one may replace the access group with the connection point group containing a number of co-located connection points.*

network level. To adapt the framework to the modelling requirements for telecom systems, a number of enhancements had to be made as briefly described below. This *enhanced RM-ODP framework* [G.851.1] provides an object-oriented framework for the modelling of distributed management systems. The OSI Management Framework was not chosen, partly due to its weakness in providing mapping to management requirements and the lack of support for distribution.

The salient features of a distributed system are described in terms of five viewpoints:

- Enterprise Viewpoint;
- Information Viewpoint;
- Computational Viewpoint;
- Engineering Viewpoint;
- Technology Viewpoint.

These viewpoints are self-contained, orthogonal specifications of a system. Additionally, certain relationships between the viewpoints need to be fulfilled to preserve the integrity of the overall system.

The main enhancement to RM-ODP consists of introducing a finer modelling granularity into the Enterprise Viewpoint to reflect the granularity of the network resource types of [G.805]. Also, all constructs of all viewpoints have been provided with unique labels for backward traceability to the functional requirements defined in the Enterprise Viewpoint. This is a fundamental mechanism for the support of conformance testing, and also for estimating the cost of implementation related to particular requirements.

Because the target application is a model for management, only the system aspects subject to management, i.e. the management requirements, need to be represented in the model. The management requirements are expressed in terms of actions with associated policies (enforcements or restrictions). This is done in the Enterprise Viewpoint and implies that the requirements become an integrated part of the model itself. Another implication is that the Enterprise Viewpoint becomes a repository for management requirements, i.e. the management specification *per se*.

3.2.1 The Enterprise Viewpoint

The *Enterprise Viewpoint* describes the open, distributed system in terms of the purpose, scope and policies of the system. It allows the invoker to express its functional requirements in terms of actions and policies on the actions to be performed by the provider, thereby establishing the contract between the invoker and the provider. [G.852.2] specifies the Enterprise Viewpoint description of a transport network resource model. A *Resource* in this context is one of the architectural components defined in [G.805] to be managed at the network level by a transport network level management service.

The resources are structured into *Enterprise Objects* that support *Actions*. The objects, or rather the roles (a role is a fraction of the object behaviour) that the objects play, interact through the actions.

The main purpose of the Enterprise Viewpoint is to identify the network resources and the associated policies necessary to fulfil the management requirements. A combined graphical/textual representation of the Enterprise Viewpoint concepts applicable to IP topology management is shown in Figure 8. The topological link is a link supported by a single trail in the server layer.

A composition of enterprise objects formed to meet a common objective is called a *Community*. The community does not reference objects – only the roles they play. The community specifies the scope of a specific management task being addressed. A community comprises a set of roles, a set of actions and a set of policies to satisfy the cooperative objective, or contract, they play.

An ordered series of actions combined to provide more comprehensive functions is called an *Activity*.

The service contract may be used for defining the *Service Level Specification* (SLS) of a *Service Level Agreement* (SLA), which expresses the agreed functionality to be used in interactions between the caller and the provider.

In addition to capturing the functional requirements, the Enterprise Viewpoint also serves as a roadmap towards the other viewpoints. Actions in the Enterprise Viewpoint map to interface operations in the Computational Viewpoint. The client and provider roles map to computational objects. Enterprise actions are normally concerned with the manipulation (create, delete, associate, etc.) of [G.805] network resource roles like subnetworks, etc. Network resource roles map to objects, attributes or relationships in the Information Viewpoint.

3.2.2 The Information Viewpoint

The static structure of a system is described in terms of *Information Objects*, *Attributes*, *Information Relationships* and *Static Invariants (static constraints)*. The information objects constitute invariants, attributes and relationships.

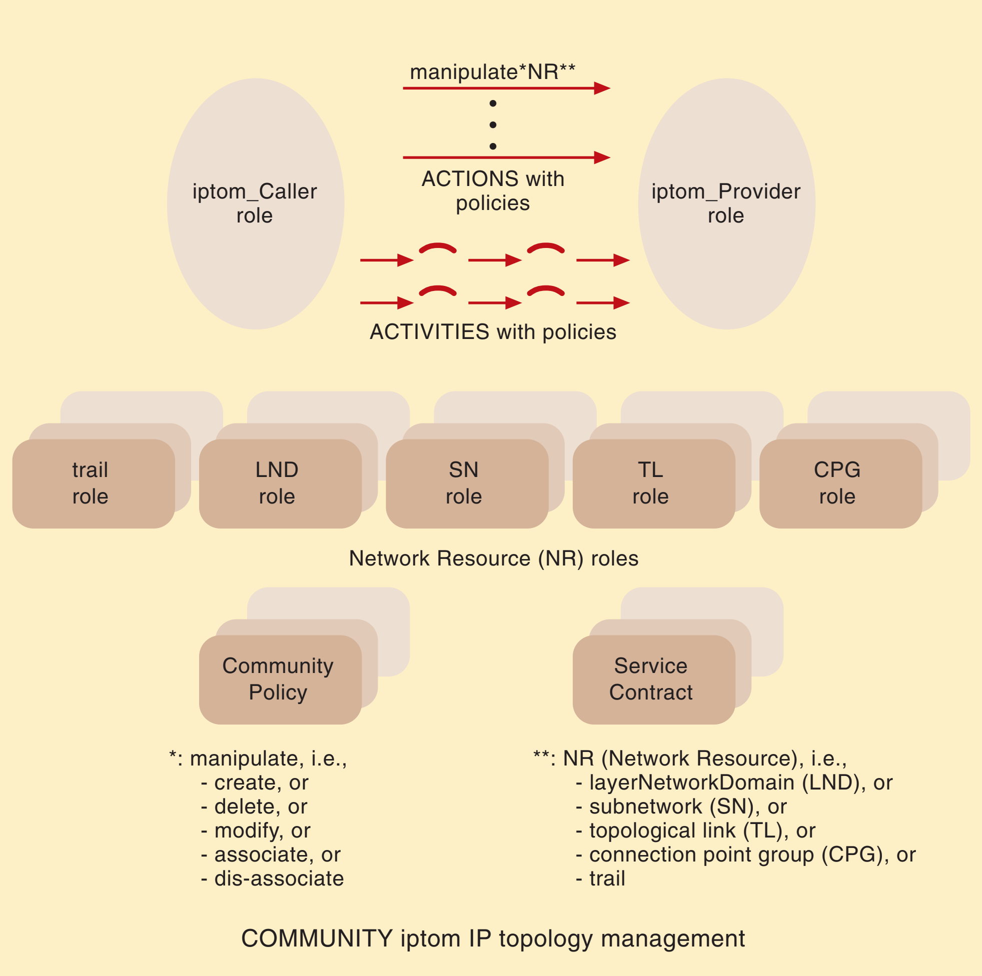


Figure 8 Enterprise Viewpoint – concepts

Community: A composition of Enterprise Objects formed to meet an objective.
Role: Part of the behavior of an Enterprise Object.
Policy: An Obligation, Permission, or Prohibition on an Action, an Activity or the Community as a whole.
Service Contract: A consistent sub-set of the functionality of an imported community.:

They may either be provided as structured English text and graphical UML diagrams or by means of the formal language Z [Zintro].

[G.853.1] is a library of information elements directly defined on the basis of the network resource types in [G.852.2]. Information elements for specific management areas are produced using these as superclasses and defining specializations with new requirements and elements.

By convention, new elements are provided with the community prefix, in the example case, iptom. Elements defined in [G.853.1] are either left unprefixed or prefixed G.853.1. The Information Viewpoint concepts are illustrated in Figure 9.

The CPG role has no direct counterpart in [G.853.1], so a new information object class, iptomCTPG has been defined.

The relationships are provided with the relationship name, the role names and the role cardinality. The arrows in Figure 9 point to the information objects playing the various relationship roles.

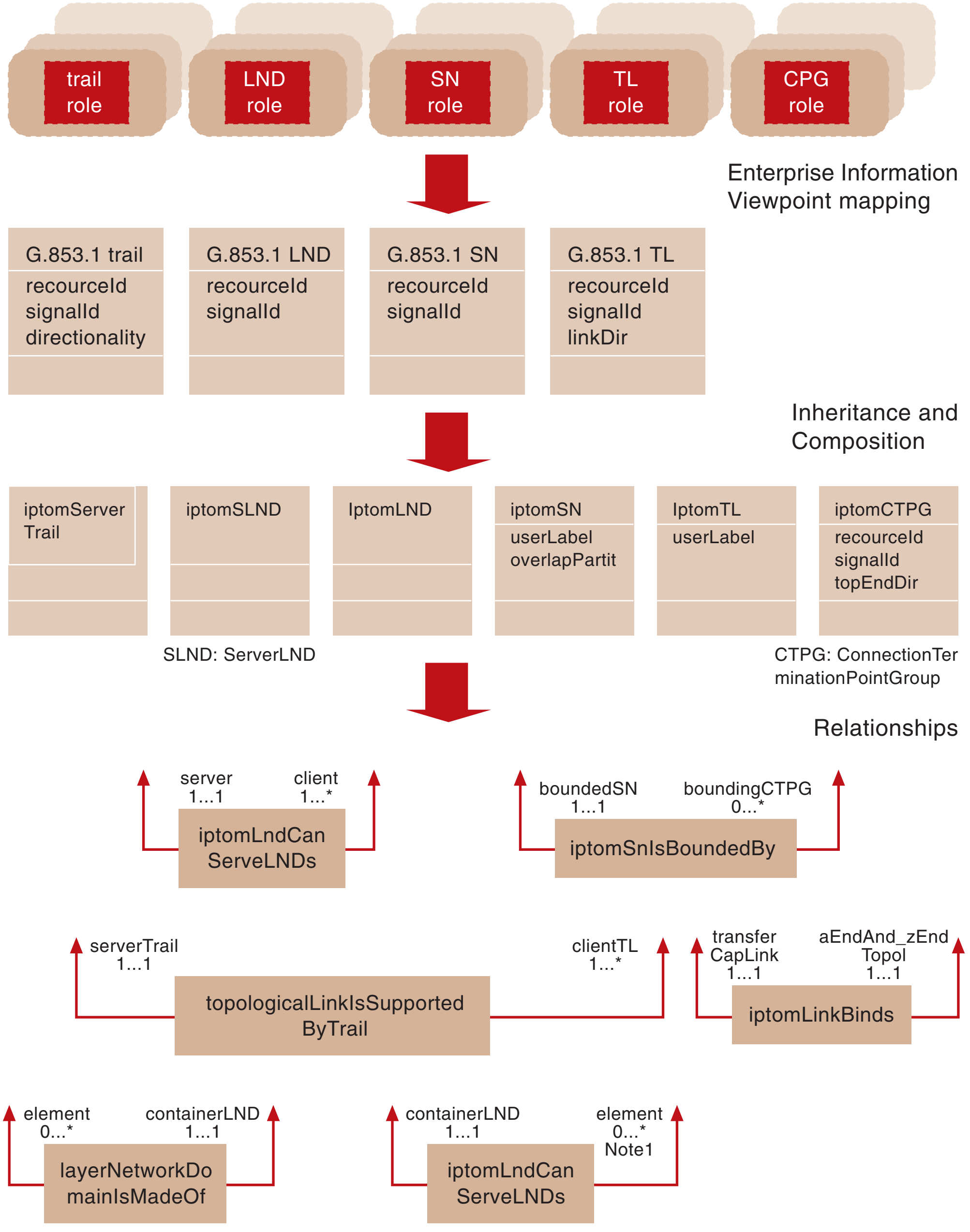
The Information Viewpoint is the ultimate source for the definition of information elements within the system. This is reflected by the Parameter Matching clause in the Computational Viewpoint which maps the input and output parameters to the corresponding information elements.

The syntax applied when defining elements in the Information Viewpoint is illustrated by examples in the following sub-sections.

3.2.2.1 Information Object: Topological Link (example)

This information type is related to the following enterprise entity:

Figure 9 Information Viewpoint – concepts



NOTE1: The iptomSN and iptomCTPG information objects are taking the “element” role in layerNetworkDomainIsMadeOf relationships not shown in the figure.

<COMMUNITY:tem, ROLE:topologicalLink>
DEFINITION

“A topologicalLink information object represents *‘a link provided by one and only one server trail, in a client layer’ (G.852.2 definition).*

This topologicalLink information object type is a subtype of the networkInformationTop information object type.”

ATTRIBUTE

signalIdentification

“The signalIdentification describes the signal that is transferred across the link.”

linkDirectionality

“The linkDirectionality attribute characterizes the ability of the associated resource to carry traffic in one, two, or undefined direction.“

INVARIANT

inv_directionality

“The linkDirectionality attribute value cannot be set to undefined.”

POTENTIAL_RELATIONSHIPS

<topologicalLinkIsSupportedByTrail>
<compoundLinkHasLinks>
<linkBinds>
<linkHasLinkConnections>
<linkIsTerminatedByLinkEnds>
<snIsPartitionedByLinks>

3.2.2.2 Information Attribute: Link Directionality (example)

DEFINITION

“The link directionality attribute characterizes the ability of the associated resource to carry traffic in one, two or undefined direction.”

STATE

undefined

“There is no indication on the ability of the resource to carry the signal in one or two directions.”

unidirectional

“The resource is able to carry the signal in only one direction from A_end to Z_end.”

bidirectional

“The resource is able to carry the signal in two directions.”

3.2.2.3 Information Relationship: Topological Link is Supported by Trail (example)

This relationship type is related to the following enterprise entity:

<COMMUNITY:tem,*ROLE:topological link,PROPERTY:adaptation*_relation>, <COMMUNITY:tem,ROLE:trail,PROPERTY: adaptation_relation>

DEFINITION

“The topologicalLinkIsSupportedByTrail relationship class describes the relationship that exists between topologicalLinks of a given layer network (known as the client layer network) and the trail that supports them in a server layer network.”

ROLE

clientTL

“Played by instances of the<topologicalLink> information object type or subtype.”

serverTrail

“Played by an instance of the <trail> information object type or subtype.”

INVARIANT

inv_serverTrailRoleCardinality

“One and only one instance of the role *serverTrail* must participate in the relationship.”

inv_clientTLRoleCardinality

“At least one instance of the *clientTL* must participate in the relationship.”

inv_directionality

“If the information object playing the role *serverTrail* is bidirectional, then the information objects playing the role *clientTL* must be bidirectional.”

inv_signalIdentification

“In a given relationship instance of topologicalLinkEndSupportedByNetworkTTP, the information object playing the role *servertrail* must have a different signalIdentification value than the information object playing the role *clientTL* as defined in Recommendation G.805 (compliant values are technologies dependent and defined in the corresponding Recommendations, e.g. G.783 for SDH).”

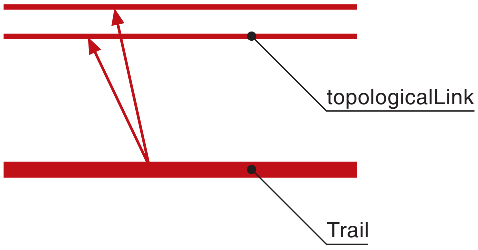


Figure 10 The topological link is supported by trail relationship

Potential relationships are not inherited in a subclass specification. Mandatory relationships are, however, inherited.

3.2.3 The Computational Viewpoint

The Computational Viewpoint describes the functional decomposition into structures suitable for distribution. The dynamic behaviour of the system is described in terms of interactions between *Computational Objects* that support *Computational Interfaces*, and *Dynamic Invariants* (dynamic constraints) on the objects and their interfaces. For each interface, a set of *Computational Operations* is defined.

Each operation is invoked providing a number of input parameters and upon successful execution, a number of output parameters are returned. Each parameter has a name, type specifier and a value assigned. Every parameter is mapped to the corresponding element in the Information Viewpoint in the Parameter Matching clause.

The computational objects may have *Operational* and *Notification Interfaces*. For each successful operation, a report operation notifying relevant external recipients is generated.

The invariant state of a system before and after the execution of an operation is specified by defining the state of the relationships and attributes supported in the form of a set of *Pre- and Post Conditions*. This is part of the “Design by contract” methodology described in [OOsw]. Whenever an invariant is violated, a specific exception is raised. For each exception, an explanatory text as well as a type specifier are provided.

The operations are mapped to the corresponding actions in the Enterprise Viewpoint.

Operational interfaces are specified in terms of *Operation Signatures* and associated *Behaviour*. The operations are described in a communication protocol neutral fashion. Protocol specific constructs for the communication protocol chosen are added in the Engineering Viewpoint. The parameters are defined in the ASN.1 language [ASN.1].

The computational objects provide the finest granularity of objects referenced in the mapping to the Engineering Viewpoint.

3.2.4 The Engineering Viewpoint

The *Engineering Viewpoint* describes the elements actually chosen for distribution and in terms that enable the introduction of a number of distribution transparencies such as location transparency, migration transparency, etc. The most salient engineering constructs are the cluster for co-locating sets of engineering objects and the capsule for the allocation of clusters to computing nodes.

The engineering viewpoint specifies operations for specific communications protocols such as CMIP and CORBA IIOP.

3.2.5 The Technology Viewpoint

The *Technology Viewpoint* describes the implementation aspects of the system. It will not be further described here.

4 Functional Model

A functional model has been defined for IP topology management in [genIPmodel]. The management requirements are defined in the enterprise viewpoint of this functional model. The requirements are expressed as a set of policies associated with the topology management community as a whole, and a set of actions applicable to the various network resources managed by the community.

4.1 Community Purpose, Roles and General Policies

The topology management community shall manage the topology of a layer network domain and the relationships between the network resources of this layer network domain. Services are offered to create and delete the following resources: subnetwork, link, topological link, and connection termination point group. These creation and deletion events may be reported to potential notification receivers by use of the offered reporting actions. Services are also offered to create and delete associations between connection termination points and connection termination point groups, and between connection termination points and subnetworks. The community supports subnetwork partitioning.

Several roles are defined in the topology management community. The network resource roles are the following: layer network domain, subnetwork, link, topological link, connection termination point, and connection termination point group. With the exception of connection termination point group, these roles represent resources as defined in [G.852.2]. The network resources represented are described in Section 3. Additional roles defined are those of the caller (i.e. the client invoking the community actions), the provider (i.e. the server performing the community actions), and the notification receiver (i.e. a receiver of the community reporting actions).

A set of general policies has been defined for the community. Some of these policies may be termed “generic”, in the sense that they apply to communities in general, not specifically to the IP topology management community. An example is “OBLIGATION viewingCapabilities”, in which the provider is obliged to support a view of resource properties and relationships sufficient for the caller to request the contracted services. Another example is “OBLIGATION serviceRejection”, in which the provider is obliged to identify the policy (obligation or prohibition) that has not been fulfilled in case of service rejection.

In addition to the generic community policies, several community-specific policies have been defined. These policies define fundamental rules and options for subnetwork partitioning.

4.2 Community Actions

This section describes the various actions offered by the topology management community. In general, all action requests will be rejected by the provider if a community policy is violated.

4.2.1 Actions Related to Subnetworks

The *create subnetwork* action creates a subnetwork inside a layer network domain. The caller may provide a unique identifier to be applied to the created subnetwork. If the provided identifier is not unique within the layer network domain, the action request will be rejected. The caller may also provide a user-friendly label for his own use. This user label need not be unique within the layer network domain. When the subnetwork has been created, the provider returns a subnetwork identifier. A list of connection termination point groups that are associated with the created subnetwork may also be returned, if this is part of the provider policy.

The *delete subnetwork* action deletes a subnetwork inside a layer network domain. A request for subnetwork deletion will be rejected if the subnetwork in question is associated with any connection termination point groups or subnetworks (i.e. in a subnetwork partitioning hierarchy).

The *associate subnetwork with subnetwork* action associates a composite subnetwork with one or more component subnetworks in a subnetwork partitioning hierarchy. The caller shall provide the identifiers of the composite subnetwork and the component subnetwork(s) to be associated. The action request is rejected if the specified component subnetworks do not belong to the same partitioning level.

The *disassociate subnetwork from subnetwork* action disassociates a composite subnetwork from one or more component subnetworks in a subnetwork partitioning hierarchy. The caller shall provide the identifiers of the composite subnetwork and the component subnetwork(s) to be disassociated. The action request is rejected if any of the specified component subnetworks are not associated with the composite subnetwork.

The *report subnetwork creation* action reports the creation of a subnetwork instance to a notification receiver. The identifier of the created subnetwork is included in the report. The report subnetwork deletion action reports the deletion of a subnetwork instance to a notification receiver. The identifier of the deleted subnetwork is included in the report.

The *report association of subnetwork with subnetwork* action reports the association of a composite subnetwork with one or more component subnetworks to a notification receiver. The identifiers of the involved subnetworks are included in the report. The report disassociation of subnetwork from subnetwork action reports the disassociation of a composite subnetwork from one or more component subnetworks to a notification receiver. The identifiers of the involved subnetworks are included in the report.

4.2.2 Actions Related to Links

The *create link* action creates a link between two connection termination point groups. The caller shall provide the two link endpoints in the action request. The provided endpoints must both be associated with subnetworks. If the provided endpoints are not acceptable, the action request will be rejected by the provider. The caller may define constraints on the directionality of the requested link. The caller may provide a unique identifier to be applied to the created link. If the provided identifier is not unique within the layer network domain, the action request will be rejected. The caller may also provide a user-friendly label for his own use. This user label need not be unique within the layer network domain. When the link has been created, the provider returns a link identifier.

The *delete link* action deletes a link inside a layer network domain. A request for link deletion will be rejected if the specified link contains link connections.

The *report link creation* action reports the creation of a link instance to a notification receiver. The identifier of the created link is included in the report. The *report link deletion* action reports the deletion of a link instance to a notification receiver. The identifier of the deleted link is included in the report.

4.2.3 Actions Related to Topological Links

The *create topological link* action creates a topological link between two connection termination point groups. The caller shall provide the two topological link endpoint in the action request. The provided endpoints must both be associated with subnetworks. If the provided endpoints are not acceptable, the action request will be rejected by the provider. The caller may define constraints on the directionality of the requested topological link. The caller may provide a unique identifier to be applied to the created topological link. If the provided identifier is not unique within the layer network domain, the action request will be rejected. The caller may also provide a user-friendly label for his own use. This user label need not be unique within the layer network domain. When the topological link has been created, the provider returns a topological link identifier.

The *delete topological link* action deletes a topological link inside a layer network domain. A request for topological link deletion will be rejected if the specified topological link is associated with a server trail.

The *report topological link creation* action reports the creation of a topological link instance to a notification receiver. The identifier of the created topological link is included in the report. The *report topological link deletion* action reports the deletion of a topological link instance to a notification receiver. The identifier of the deleted topological link is included in the report.

4.2.4 Actions Related to Connection Termination Points and Connection Termination Point Groups

The *create connection termination point group* action creates a connection termination point group inside a layer network domain. The caller may define constraints on the directionality of the connection termination points that may be associated with the requested connection termination point group. The caller may provide a unique identifier to be applied to the created connection termination point group. If the provided identifier is not unique within the layer network domain, the action request will be rejected. The caller may also provide a user-friendly label for his own use. This user label need not be unique within the layer network domain. When the connection termination point group has been created, the provider returns a connection termination point group identifier.

The *delete connection termination point group* action deletes a connection termination point group inside a layer network domain. The action request will be rejected if the specified connection termination point group is associated with any connection termination points, or if it terminates a link or topological link.

The *associate connection termination point with connection termination point group* action creates an association between a connection termination point and a connection termination point group. The caller shall identify the entities to be associated in the action request. The provider will reject the action request if the specified connection termination point is already associated with a connection termination point group, or if the directionality of the specified entities is not compatible.

The *disassociate connection termination point from connection termination point group* action deletes an association between a connection termination point and a connection termination point group. The caller shall identify the entities to be disassociated in the action request.

The *associate connection termination point group with subnetwork* action creates an association between a connection termination point group and a subnetwork. A connection termination point group may be associated with one or more subnetworks, depending on the levels of partitioning supported. This action makes the connection termination point group available for routing across the subnetwork. The caller shall identify the entities to be associated in the action request. The provider will reject the action request if the specified connection termination point group is already associated with the specified subnetwork.

The *disassociate connection termination point group from subnetwork* action deletes an association between a connection termination point group and a subnetwork. The caller shall identify the entities to be disassociated in the action request. The provider will reject the action request if the specified connection termination point group is not associated with the specified subnetwork, or if the specified connection termination point group terminates a link or topological link.

All the actions described above have a corresponding *report* action that reports the event in question to a notification receiver. In all cases, the identifiers of all involved entities are included in the report.

5 Example: Delete Topological Link

This chapter presents the enterprise, information and computational viewpoints for the *delete topological link* action as an example of the modelling methodology. As pointed out in Section 2.3, a topological link is a link supported by a single trail in the server layer. In an MPLS domain, for example, this implies that for an IP link like the one in Figure 3 to be classified as a topological link, it has to be supported by a single LSP.

The *delete topological link* action deletes a topological link inside a layer network domain. A topological link may not be deleted if a server trail is assigned to it.

5.1 Enterprise Viewpoint

The enterprise viewpoint defines the policy of the *delete topological link* action in terms of four OBLIGATIONS. The model text is as follows:

Delete topological link
"This action deletes a topological link inside a layer network domain. No other resource is deleted by this action."

ACTION_POLICY

OBLIGATION inputTopologicalLinkId
"The caller shall provide the identifier of the topological link to be deleted."

OBLIGATION noExistingTopologicalLink
"This action will fail if the topological link specified does not exist within the layer network domain. In the case of failure, the provider shall return the identifier in error."

OBLIGATION noServerTrail
"This action will fail if a server trail is still assigned to the topological link specified."

OBLIGATION successTopologicalLinkDeleted
"When the action is successful, the provider shall indicate this to the caller."

As stated by the action policy, the caller shall provide the identifier of the topological link he wants to delete (OBLIGATION inputTopologicalLinkId). The provider will of course reject the action request if the specified link does not exist (OBLIGATION noExistingTopologicalLink). Additionally, the action request will be rejected if the topological link in question has a server trail assigned to it (OBLIGATION noServerTrail). An eventual successful output of the action is indicated to the caller (OBLIGATION successTopologicalLinkDeleted).

5.2 Information Viewpoint

The information viewpoint defines the information object types involved in the *delete topological link* action and the relationships that exist between object instances.

The information object types affected by the said action are iptomLayerNetworkDomain, iptomTopologicalLink, iptomServerTrail, and iptomServerLayerNetworkDomain. Figure 11 presents the inheritance diagram for these information object types.

All types inherit from the [G.853.1] networkInformationTop information object type. All information objects inherit the *resourceId* attribute from networkInformationTop. This attribute represents the unique identification of a resource, and the *resourceId* associated with an information object must be unique for its associated class.

[G.853.1] object types topologicalLink, transportConnection, and layerNetworkDomain are supertypes from which the relevant specialisations are made. A topologicalLink information object represents a link provided by one and only one server trail, in a client layer. The formal definition of the topologicalLink information object type is presented in section 3.2.2. A transportConnection information object represents a [G.805] connection, or a [G.805] trail. The transportConnection subtype trail, from which the iptomServerTrail is derived, represents a [G.805] trail. Finally, a layerNetworkDomain information object represents an administrative domain in which all resources pertain to the same [G.805] layer.

All involved subtypes inherit the *signalidentification* attribute from their respective supertypes. This attribute represents the specific format of signal that a resource carries. The specific formats are technology-specific, and are defined in technology-specific extensions. IP-specific formats have not been defined.

Additionally, iptomTopologicalLink inherits the *linkDirectionality* attribute. This attribute characterises the ability of the topological link to carry traffic in one, two or undefined direction. The formal definition of the *linkDirectionality* attribute is presented in section 3.2.2.The standard *userLabel* attribute, representing a user-friendly label given to a resource by a user, is also present in iptomTopologicalLink information objects.

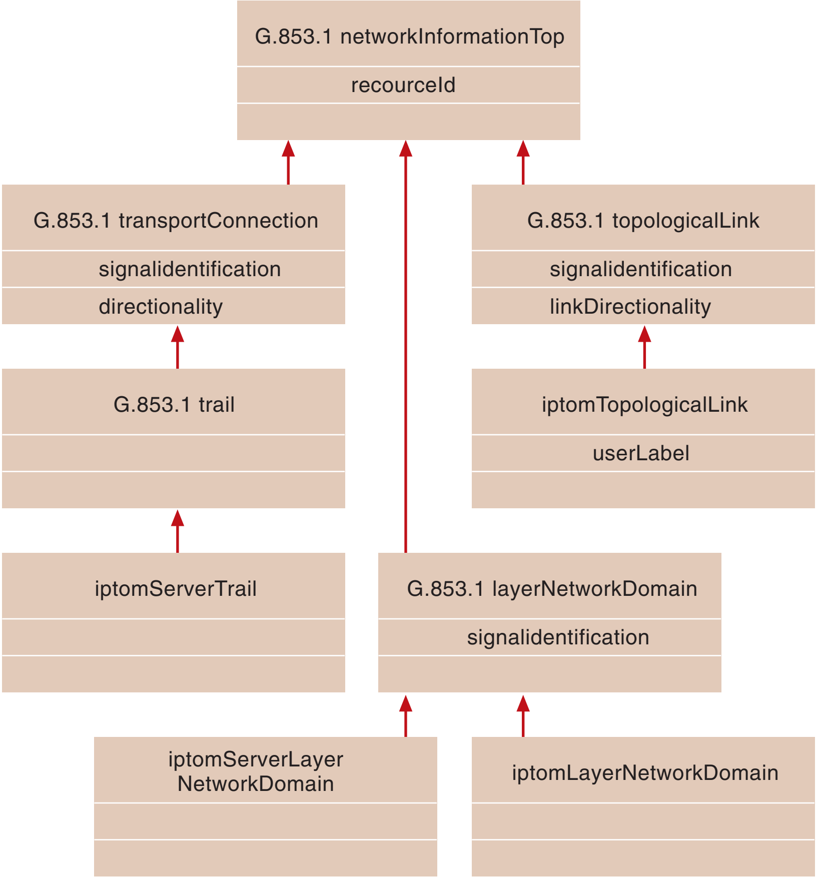


Figure 11 Inheritance diagram for topological link and related entities

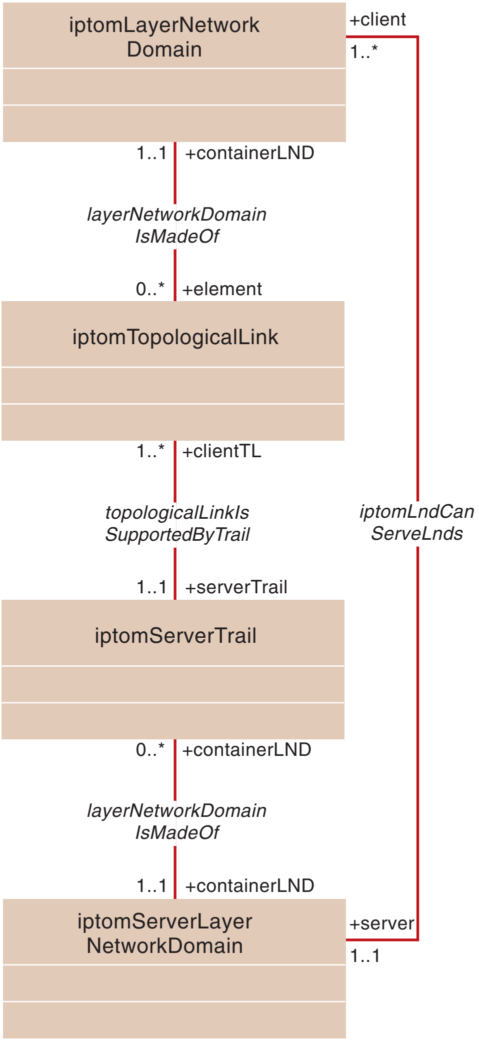
Figure 12 presents the relationship diagram for the same information object types. The diagram is restricted to the relationship types of relevance to the delete topological link action.

The *topologicalLinkIsSupportedByTrail* relationship between the iptomTopologicalLink and iptomServerTrail information objects is of special interest to the *delete topological link* action. If the topological link specified in a *delete topological link* action participates in such a relationship, it cannot be deleted, and the action request is rejected. The formal definition of the *topologicalLinkIsSupportedByTrail* relationship type is presented in section 3.2.2.

A layer network domain has a container-element sort of relationship with the network resource objects that compose it, formally represented by the *layerNetworkDomainIsMadeOf* relationship type. A *layerNetworkDomainIsMadeOf* relationship instance may include several element role instances, but it may include one and only one container LND role instance.

While the *topologicalLinkIsSupportedByTrail* relationship type represents a relationship between a server layer trail and one or more client layer topological links, the *iptomLndCanServeLnds* relationship type represents the corresponding relationship between server and client layer network domains.

Figure 12 Relationship diagram for topological link and related entities



5.3 Computational Viewpoint

The computational viewpoint specifies the resource provisioning interface for the *deleteTopologicalLink* action as follows:

```
<COMMUNITY: IP topology management,
ACTION: delete topological link>
OPERATION deleteTopologicalLink {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        topologicalLink: TopologicalLinkId;
    OUTPUT_PARAMETERS
        -- none
    RAISED_EXCEPTIONS
        incorrectTopologicalLink:
            TopologicalLink;
        serverTrailExisting: NULL;
        failureToDeleteTopologicalLink: NULL;
BEHAVIOUR
    SEMI_FORMAL
        PARAMETER_MATCHING
            layerND:
                <INFORMATION OBJECT:
    iptomLayerNetworkDomain>;
            topologicalLink:
            <INFORMATION OBJECT:
            iptomTopologicalLink>;
        PRE_CONDITIONS;
    inv_existingTopologicalLink
        "topologicalLink refers to the element
        in the <layerNetworkDomainIsMadeOf>
        relationship where layerND refers to
        containerLND."
    inv_noServerTrail
        "topologicalLink shall not refer to
        any clientTL of a <topologicalLinkIs
        SupportedByTrail> relationship."
        POST_CONDITIONS
    inv_noTopologicalLink
        "topologicalLink does not participate in
        any <layerNetworkDomainIsMadeOf>
        AND <iptomLinkBinds> relationships."
    EXCEPTIONS
        IF PRE_CONDITION inv_existingTopologicalLink NOT_VERIFIED
            RAISE_EXCEPTION incorrectTopologicalLink;
        IF PRE_CONDITION inv_noServerTrail NOT_VERIFIED
            RAISE_EXCEPTION serverTrailExisting;
        IF POST_CONDITION inv_no
        TopologicalLink NOT_VERIFIED
            RAISE_EXCEPTION failureToDeleteTopologicalLink;
;}
```

INPUT_PARAMETERS specifies the input parameters to this action. The topologicalLink parameter identifies the topological link that is to be deleted. The layerND parameter is implicitly provided, as it is the containing entity of the topological link (see Figure 12).

OUTPUT_PARAMETERS specifies the output parameters from this action, which is normally none.

RAISED_EXCEPTIONS specifies return values in case the action fails.

PRE_CONDITIONS specifies the conditions that have to be present prior to the action request. The invariant inv_existingTopologicalLink refers to the fact that the topological link has to exist. The invariant inv_noServerTrail refers to the requirement that the topological link to be deleted shall not be associated with a server trail.

POST_CONDITIONS specifies the conditions that are present when the action is completed. The invariant inv_noTopologicalLink refers to the fact that the deleted topological link does not exist after the *deleteTopologicalLink* has successfully completed.

EXCEPTIONS specifies the exceptions that may occur if any of the pre- or postconditions described above are not fulfilled, with reference to the RAISED_EXCEPTIONS clause.

6 Conclusion

A protocol neutral network level model addressing the topological aspects of IP networks on the basis of the architectural components subnetwork, link and connection point group has been developed. This paper starts out analysing two QoS-aware network architectures. It carries on to provide an informal description of the topology model, and also presents an abstract traffic model for DiffServ-over-MPLS networks. The modelling methodology, an enhanced version of the Reference Model for Open Distributed Processing, is briefly described in Section 3. The functionality of the formal model is described next and finally, as an example, the Enterprise, Information and Computational Viewpoint specifications for the deletion of a topological link are presented.

This model is most useful in representing the geographical distribution of IP network resources, partly to support the dissemination of topological information in QoS-aware networks. As previously mentioned, a number of functional elements such as queueing, scheduling and shaping are candidates for inclusion in a more detailed version of the model.

A model for setting up LSP tunnels in DiffServ-over-MPLS networks via a management interface has been developed already [genIPmodel] providing the basis for the server layer support of the IP layer in a layered model. Extensions to accommodate VPNs and Multicast will be considered.

7 References

[ASN.1] ITU-T. 1994. *Abstract Syntax Notation One (ASN.1): Specification of Basic Notation.* Geneva 07/94. (ITU-T rec.X.680.)

[G.805] ITU-T. 1995. *Generic functional architecture of transport networks.* Geneva 11/95. (ITU-T rec. G.805.)

[G.851.1] ITU-T. 1999. *Management of the transport network – Application of the RM-ODP framework.* Geneva 03/99. (ITU-T rec. G.851.1.)

[G.852.2] ITU-T. 1999. *Management of transport network – Enterprise viewpoint description of transport network resource model.* Geneva, 03/99. (ITU-T rec. G.852.2.)

[G.853.1] ITU-T. 1999. *Management of transport network – Common elements of the information viewpoint for the management of a transport network*. Geneva, 03/99. (ITU-T rec. G.853.1.)

[GenIPmodel] Henriksen, T et al. 2000. *A generic model for IP based networks.* Kjeller, Telenor R&D. (R&D report R 39/2000.)

[ITUmeth] Henriksen, T. 2001. Network level modeling in ITU. *Telektronikk*, 97 (1), 147–155.

[OOsw] Meyer, B. 1997. *Object oriented software construction.* Upper Saddle River, NJ, Prentice Hall.

[PASTE] IETF. 1998. *A provider architecture for differentiated services and traffic engineering (PASTE)*. (RFC2430.)

[RessHandlIP] Svinnset, I et al. 2001. *Resource handling in IP networks.* Kjeller, Telenor R&D. (R&D report R 5/2001).

[TE_REQ] IETF. 1999. *Requirements for traffic engineering over MPLS.* (RFC2702.)

[X.901] ITU-T. *Basic reference model for Open Distributed Processing – Part 1: Overview.* Geneva. (ITU-T rec. X.901.)

[X.902] ITU-T. *Basic reference model for Open Distributed Processing – Part2: Foundations.* Geneva. (ITU-T rec. X.902.)

[X.903] ITU-T. *Basic reference model for Open Distributed Processing – Part 3: Architecture.* Geneva. (ITU-T rec. X.903.)

[X.904] ITU-T. *Basic reference model for Open Distributed Processing – Part 4: Architectural Semantics.* Geneva. (ITU-T rec. X.904.)

[Zintro] Potter, B et al. 1992. *An introduction to formal specification and Z.* New York, Prentice Hall.

Traffic Measurements in IP Networks

BRYNJAR Å. VIKEN AND PEDER J. EMSTAD

Brynjar Å. Viken (31) is Research Scientist at Telenor R&D, Trondheim. His research interests are performance measurements and analysis of communication networks with a special interest in IP networks. He is currently pursuing a PhD at the Norwegian University of Science and Technology.

brynjar-age.viken@telenor.com

Dr.Ing. Peder J. Emstad (61) is Professor at the Department of Telematics at the Norwegian University of Science and Technology. His research interests are performance modelling and analysis of communication networks and switching systems.

peder@item.ntnu.no

There is an increasing need for traffic performance measurements in IP networks. The Internet has become an important part of the communication infrastructure, and network operators and service providers need to measure the flow of traffic. Today's best effort networks give variable quality to users. This may be alleviated in the near future offering differentiated services to the users. This article presents an overview of issues related to traffic measurements in IP networks, including characterization of services, measures of interest, measurement methods, measurement infrastructures and post-processing of measurement data. The main focus is on network-level performance measurements from the perspective of a network operator. A novel model appropriate for precise and concise discussion and definition of network level measurement issues is introduced and used to precisely describe network-level performance metrics. A discussion of the implications on traffic measurements by introducing means and mechanisms for service differentiation concludes the paper.

1 Introduction

The traditional Internet provides an unreliable best-effort packet delivery. Packets can be lost, errored, duplicated and delivered out-of-order. The packet delivery is characterized by the absence of service guarantees and fairness. New emerging applications with the need for end-to-end performance guarantees drive the introduction of service differentiation in IP networks. Examples of these applications are IP telephony, network games, audio and video streaming. The introduction of service differentiation in IP networks increases the importance of traffic measurements. Traffic measurements are vital for several areas including network management, traffic engineering, charging and billing, and monitoring of service level agreements.

1.1 Actors Involved in Traffic Measurements

Actors interested in collecting measurement data are end-users, network providers, service and content providers, vendors and researchers. The respective actors have various perspectives and needs of traffic measurements on various time-scales. The time-scales may be classified as short (seconds – minutes – hours), medium (days – weeks) and long (months – years).

End-users need to gather measurement data to ensure that the IP services they receive from their service providers meet the agreed levels of service. In the coming years as service differentiation is introduced, traffic measurements will play a major part in evaluating service quality versus price for services an end-user receives. The main focus of the end-user is end-to-end performance of certain services where the network is considered as a black box.

Network operators are focused on the network domain under their management. Measurements are important for the network operators both on a day-to-day basis and for long term planning and engineering. A network operator needs to perform daily measurements to diagnose network problems before they occur, perform troubleshooting and solve network failures. Historical data form a foundation on which decisions can be made. That is, planning, optimizing and traffic engineering of the network domain. An operator must also monitor performance to ensure that the services delivered to peering networks, service and content providers, and end-users meet the service level agreements. A network operator will need information from peering networks in order to decide how to route traffic of various service qualities. An end-to-end path often crosses several network domains and the end-to-end performance is beyond the control of one individual network operator. Evidently, efforts to both deliver and measure end-to-end services require co-operation between network operators, service providers and content providers.

Service and content providers must rely on other actors to provide end-to-end network services for their customers. Thus, it is crucial for service and content providers that their service level agreements with other actors are fulfilled and that the end-to-end services delivered satisfy the QoS requirements of their customers. Service and content providers can collect some measurement data themselves, e.g. from service specific equipment (Video on demand servers, VoIP gateways, etc.), but depend on network providers to measure network performance.

Table 1.1 Perspectives and needs for measurements on various time scales

	Short term	Medium term	Long term
End-user Service and content provider	Monitor SLAs[1]	Evaluate SLAs	Select appropriate providers
Network operator	Monitor SLAs Detect and solve problems in the network	Network configuration Routing of packets	Select peering networks Replace nodes and increase capacity of links Change network architecture and introduce new technology
Vendor	Test components	Large scale tests	Develop new products
Researcher			Models for analysis and simulations Design new solutions

[1] *SLA – service level agreement*

Performance requirements and traffic characteristics are important input parameters for the design of new network components and concepts. Vendors perform tests of single components (router, switch, etc.). Testbeds must also be established to perform large-scale tests of e.g. new network mechanisms and network architectures. Such testbeds have no traffic load so a realistic traffic load must be generated. There is also a need for accurate traffic measurements to derive performance measures. [Heegaard] presents such a distributed test environment for IP networks.

Finally, measurement data is important as input to researchers to increase the understanding of IP networks and develop better solutions. Researchers depend on accurate measurement data as an input to make realistic predictions and estimates and to build models for analysis and simulations. For the researcher, the measurements of today's Internet support the improvement of existing technology and the development of new technologies.

Table 1.1 summarizes the need for measurements for various actors on different time scales.

1.2 Collecting and Analyzing Traffic Measurements

The measurement process can be divided into several phases. First, the measurements are planned based on the need of an actor. That is, the measurements should be tailor-made for observing specific parameters. Second, the measurement infrastructure to collect and analyses measured data is designed, developed and tested. Third, the gathering of measurement data is performed. Finally, the collected measurement data is post-processed and analyzed. Figure 1.1 illustrates the various phases of the measurement process.

In each phase the costs of measurements can be grouped in measurement specific equipment (hardware and software), network resources and human resources. Measurement specific equipment includes hardware (processing power, primary and secondary memory) and software that is required to collect, handle and post-process measurement data. The measurement functionality can either be implemented on stand-alone devices and/or integrated into the functionality of the network nodes. Network resources include bandwidth needed to collect and transport the measurement data. Obviously, human resources are needed in every phase of the measurement process.

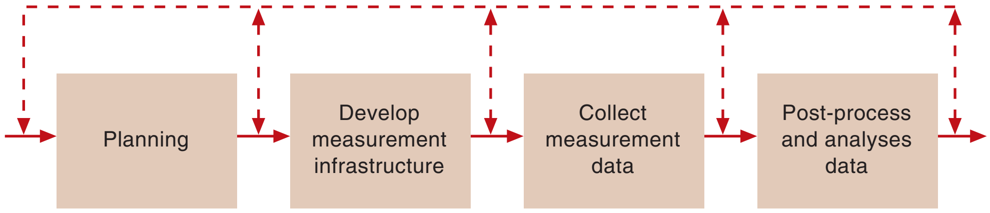

Planning
Develop measurement infrastructure
Collect measurement data
Post-process and analyses data

Figure 1.1 Phases of the measurement process

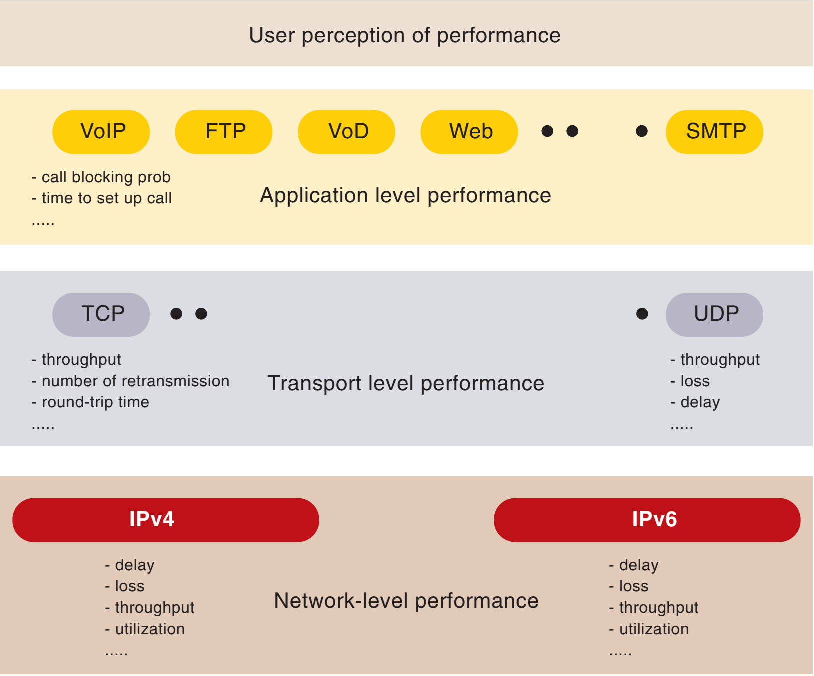


Figure 2.1 Various levels of performance

2 Measures of Interest

Network services offered have various traffic characteristics and QoS requirements. Real-time services (e.g. voice) have strict delay and loss requirements while non real-time services (e.g. file transfer) have other performance requirements (e.g. high throughput). Thus, a full service IP network carries services with a wide range of performance requirements.

A set of performance parameters is required to characterize the network performance experienced at each protocol layer, as illustrated in Figure 2.1.

Network-level measurements focus on the basic components of a network domain, namely nodes and links, and the packets traversing this communication infrastructure. Network-level measurements is a common building block from which the performance of every service carried in an IP network depends. The fundamental performance parameters for any resource sharing system are delay, throughput, loss and resource utilization.

At the *transport* level the performance of the transport protocols is addressed. The dominating transport protocols in IP networks are currently TCP and UDP. Examples of transport level metrics for TCP are number of retransmissions, the time to set up and close a TCP connection, average segment size and throughput. The relations between the network and transport level performance metrics depend on the behavior of the actual transport protocol.

At the *application* level, the performance as observed by various applications is studied. Examples of applications used in the Internet are WWW, FTP, TELNET, VoIP, video and audio streaming. Application level performance metrics for file transfers are for instance the time to establish a connection, mean file size and mean download time.

At the *user* level the human perception of a given service or system is important. Obviously, human perception depends on a wide range of factors that are difficult to measure. Ultimately, the human perception of a given service determines the success and acceptance of the service. This perception is again closely associated with expectation and costs.

The performance parameters for the transport and application layer are determined by the characteristics of actual protocols and services, e.g. for VoIP the performance parameters include the time to set up a call and the call blocking probability.

The performance measured at a higher protocol layer depends on the performance of lower layers [2)]. However, the relationship between performance measures at different protocol layers is not straightforward. Apparently, the mapping from network level measurements of delay, loss and throughput to human perception is very difficult and depends on many aspects.

This article addresses traffic measurements. However, note that dependability measures can in some cases be derived from performance measures, e.g. the service availability can be defined as the state where the end-to-end delay is less than a specified limit, and the measurement accumulates the fraction of the time the system is in this state.

Obviously, measuring the performance of various protocol layers is vital for IP traffic engineering. The focus of this article is on network-level performance measurements from the perspective of a network operator. However, the remainder of the article is also applicable to performance measurements at the transport protocol and application layers.

[2)] *Performance can also be measured at lower protocol layers than the IP layer, e.g. MAC and physical layer.*

3 Measurement Methods and Infrastructures

3.1 Introduction

The specification of an actual measurement and monitoring platform depends on what performance parameters that are to be observed. The measurement platform is characterized by the following properties:

A) Measurement points – at which points in the network it is measured (end user, interface card in end user equipment, routers, switches, servers, access network, edge routers, etc.).

B) Measurement method – active (intrusive) or passive (non-intrusive).

C) Handling and post-processing of measurements – data reduction techniques (accumulation of statistics, selection of packets or flows), accuracy in measurements (observation period, correlation, other).

D) Measurement period – time interval over which the measurements are collected (time of day, continuously or by sampling, etc.).

Systematic gathering of measurement data from a communication infrastructure requires the establishment and deployment of a measurement infrastructure. The measurement infrastructure, as illustrated in Figure 3.1, consists of a set of measurement units[3] placed at strategic locations (measurement points) in the network domain. The measurement method dictates the actual implementation of the measurement infrastructure. Today several projects develop measurement infrastructures to collect measurement data from the Internet, examples of such efforts are [NAI], [McGregor], [NIMI], [RIPE] and [Surveyor].

The two principal methods to collect network-level measurements are; either actively by inserting probe packets [RFC792] [AMP] [RIPE] [Surveyor] or passively by observing real packets [Claffy97] [Brownlee] [Cflowd] [NetFlow99] [Careces91] [Claffy98] [Viken99]. Active and passive measurements are discussed further in Section 3.3 and Section 3.4, respectively.

Collected measurement data is subject to post-processing and analyses. There are several options regarding the post-processing and analyses of measurement data, as to which data reduction techniques to use and where the post-processing units (local vs. central processing of data) should be located. The various approaches have different resource needs and offer different measurement accuracy.

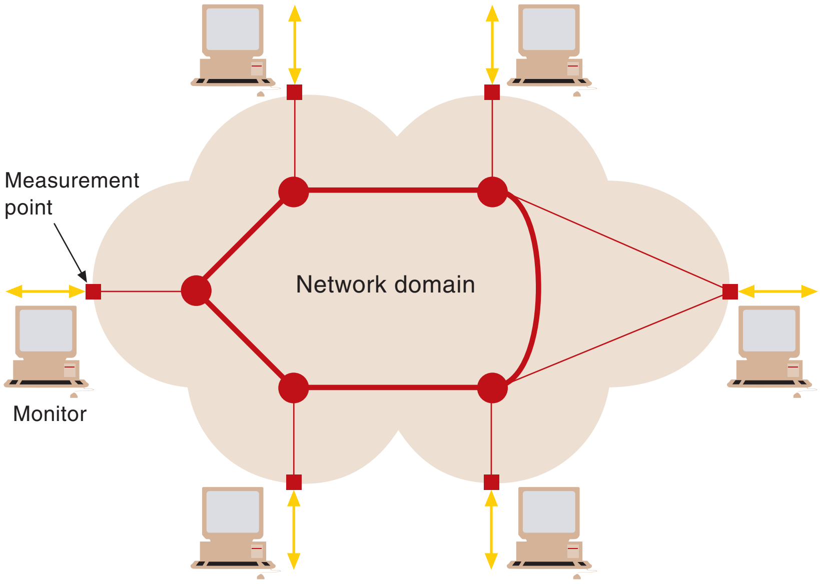


Figure 3.1 Example of a measurement infrastructure

3.2 Timestamp Requirements

Active measurements add a timestamp to every probe packet sent while passive measurements associate a timestamp with the observation of a packet. Thus, obtaining accurate timing information is crucial to minimize the measurement error for both active and passive delay measurements. Accurate estimation of one-way delay requires the clocks of the measurement units to be synchronised, accurate and have high precision. These issues are discussed in e.g. [RFC2330] [Pasztor].

The clocks of the distributed PCs collecting the measurement data can be synchronised by using GPS receivers [Surveyor] [RIPE] or the network time protocol (NTP) [RFC1305]. NTP can only provide timestamp accuracy in the range of milliseconds while the accuracy of the output signal from modern GPS receivers is well below one microsecond.

Further, generating timestamps in software introduces an additional measurement error due to operating system scheduling. Different operating systems give different measurement errors [Pasztor]. To achieve very accurate timestamps, the measurement units must be synchronized by GPS receivers and the timestamps have to be

[3] *A measurement unit is not necessarily a dedicated physical unit. The measurement functionality can be integrated in the hardware or software of a node.*

generated by specialized hardware. This solution offers accuracy in the range of a few microseconds [Pasztor] [Fraleigh].

3.3 Active Measurements

Active measurements at the network-level are carried out by inserting probe packets and observing these probe packets. The major assumption behind this approach is that the performance of probe packets is representative of the performance experienced by real packets. To measure the performance of a service, the traffic characteristics should be considered (interarrival time of packets, packet lengths, etc.). Note that active measurements are intrusive and therefore the measurements affect the system being measured.

Measurements of round-trip times and packet loss in IP networks are usually performed by software based on the ICMP Echo Reply/ Request messages [RFC792] (ping) running on PCs. The advantage of this approach is that most implementations of TCP/IP support ICMP Echo messages. Thus, measurements can be performed by pings to almost any host in the network without making any special arrangements beforehand. This approach only requires the remote host to respond to ICMP "ECHO request" messages [RFC792]. Since ICMP uses IP services, it measures network-level performance rather than transport level performance. The major drawback is that ICMP Echo Reply/ Request measurements normally are limited to measure bi-directional delays. Other disadvantages by using ping are that the routers along the path may treat ICMP packets differently from other IP packets, block or limit the rate of ICMP packets (Firewalls, etc.).

3.3.1 Active Measurements of Unidirectional Network-level Performance Parameters

The Surveyor and Ripe test traffic projects measure unidirectional performance by injecting probe packets. The dedicated measurement PCs are synchronized by using GPS receivers. The general principle is as follows; to measure the performance between a given source and destination pair, the source injects probe packets addressed to the destination. The sender adds a timestamp and sequence number to every probe packet. That is, "one-way pinging" and throughput tests are performed using dedicated measurement units located at selected measurement points. The injected probe packets can be either in-service[4] or out-of-service. That is, one can either insert dedicated test flows or insert test packets inside user flows. [Lindh] proposes a measurement infrastructure embedded in network nodes that is based on inserting special Operations and Maintenance (OAM) packets into the user traffic. This method is similar to the OAM cells used in ATM networks [Prycker95]. The measurement units must have capabilities to post-process, store, and export the collected measurement data.

Table 3.1 shows the fundamental network-level performance metrics and the suitability of the active method. It may be noted that unlike passive measurements the active measurements cannot collect detailed information about properties of user generated packets, e.g. traffic mixture, packet length and network traffic matrix.

3.4 Passive Measurements

Passive measurement data is collected by observing real packets at selected measurement

Network-level metric	Characterization of active measurements
Unidirectional delay **Unidirectional packet loss**	+ Straightforward to implement. + Easy to aggregate performance metrics at a single measurement unit. – Are the measurements representative of the performance real packets experience? – What should the traffic pattern of probe packets be? What packet interarrival time and packet lengths are representative for various service classes? – The injected probe packets disturb real traffic (Heissenberg bug).
Throughput **Link utilization**	– How can representative measurements of throughput/utilization be performed? – Active measurements generally disturb the operation of the network. Thus, lightweight tests are needed.

Table 3.1 Active measurements of performance metrics

[4] *The method requires further study in the context of IP networks.*

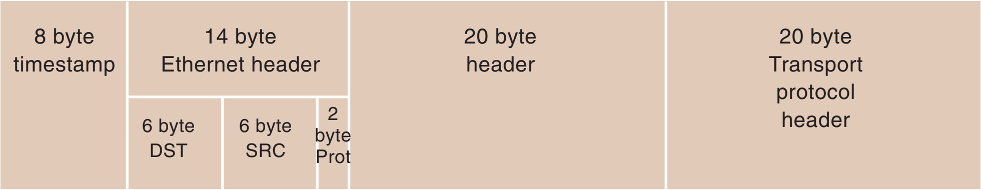


Figure 3.2 Example of packet trace data format (Ethernet)

points. As opposed to active measurements, this approach is non-intrusive and ideally the measurement process does not disturb the operation of the network. Measurement data of highly variable granularity is gathered: ranging from detailed packet traces[5] and flow records[6] to routing tables and counters from network nodes (e.g. SNMP counters on router interfaces). Packet traces and flow records require the various fields of the packet headers to be monitored while interface counters typically accumulate the number of packets and bytes transferred/dropped. The major drawback of collecting raw packet traces or flow records from high capacity networks is that huge data volumes are created. Thus, it may not be feasible to store raw data for long periods of time without performing some data reduction. Note that packet traces and flow records contain sensitive information that must be handled with care.

Examples of functionality to process, store and export passive measurement data integrated in the hardware and software of network nodes are interface counters and Cisco's NetFlow [NetFlow99] data export. The design of router architectures capable of collecting passive measurements is beyond the scope of this discussion. However, the routers should be built to collect the necessary measurement data without any disturbance to the packet forwarding capability of the router.

Specialized stand-alone PCs that collect passive measurements from high capacity links without impacting network operation are available. These passive stand-alone measurement units usually run specialized software on a hardware platform that taps information from packets traversing the link being monitored. Examples of such dedicated measurement units are Netramet [Brownlee], the OCXmon/Coral monitor [Coral] [MOAT] and the DAG monitor [Graham]. Packet traces can also be collected on regular workstations by using the tcpdump application [Tcpdump].

Each packet record carries a number of attributes that characterize the packet and the corresponding events. The attributes of a packet can be classified as endogen and exogen attributes. Endogen attributes are carried in the packet headers and user data. Exogen attributes are not carried inside the packet but are implicitly derived. Examples of exogen attributes are incoming and outgoing interface for the packet at a certain router and the time of arrival of the packet to a given node. Packet traces can contain a copy of the entire packet including headers as well as user data. However, to reduce the sensitivity and data volume usually only the IP header and transport protocol header is kept, as illustrated in Figure 3.2.

3.4.1 Passive Measurements of Unidirectional Performance Parameters Using Packet Traces

Passive measurements of unidirectional delay and loss require raw packet traces to be captured at several measurement points, as illustrated in Figure 3.3. One example of measurements collected using this method is [Graham].

Note that timestamps could be added to real packets inside the network for the purpose of passive measurements of unidirectional delay. Hence, this would allow unidirectional delay to be estimated from a single packet trace. This concept needs further study and requires specialized equipment to be developed and installed in the network. Further, for packets that contain sequence numbers it could be possible to estimate loss from a single packet trace. However, in the following it is assumed that delay and loss must be estimated by correlating the information from several packet traces.

3.4.2 Single Packet Trace

From a single trace captured at a given measurement point it is possible to compute e.g. the number of bytes sent and received to/from various remote machines, interarrival times, packet

[5] *A packet trace contains detailed information (timestamp and packet attributes) about packets observed at a certain measurement point.*

[6] *Flow records have detailed information about network flows as observed at a given measurement point. A network flow is a sequence of packets satisfying certain conditions, e.g. a unidirectional stream of packets from a specified source to a certain destination satisfying a given time-out value.*

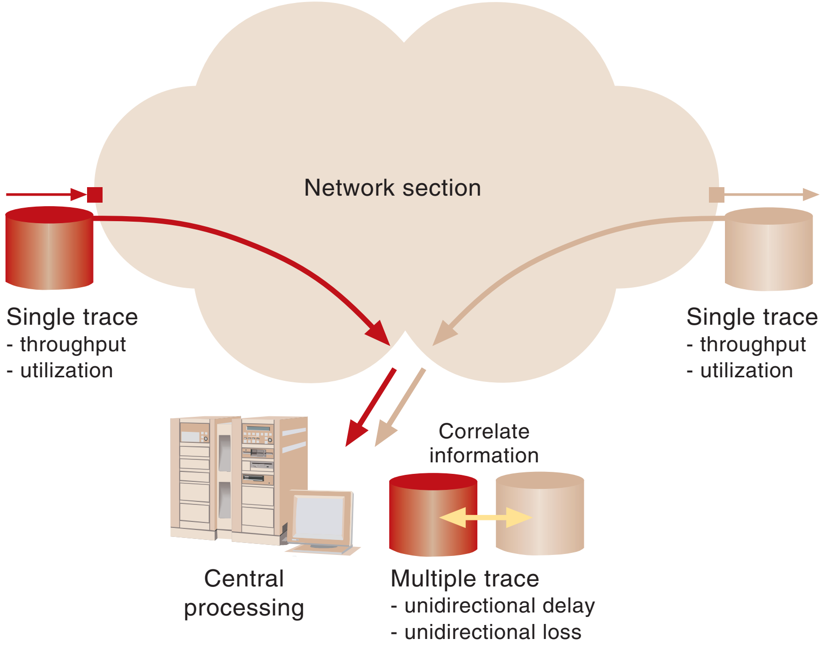


Figure 3.3 Passive measurements

Network-level metric	Passive measurement properties
Delay **Loss**	+ Measure performance as experienced by real packets. + Do not disturb the operation of the network. – Resource intensive (huge data volumes). This can be handled by using data reduction techniques.
Throughput **Utilization**	+ Measure performance as experienced by real packets. + Do not disturb the operation of the network. + Estimated from a single packet trace.

Table 3.2 Passive measurements of performance metrics

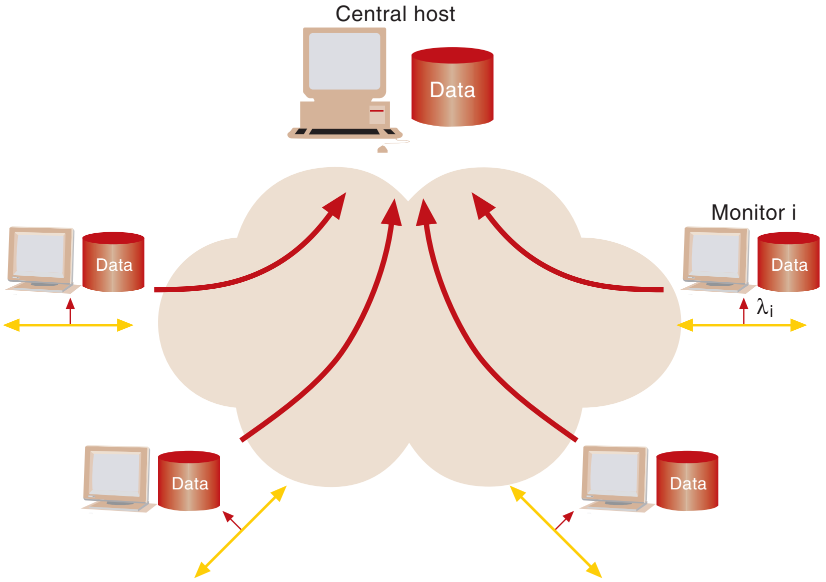


Figure 3.4 Data flow and storage for passive measurements

length distributions and link utilization. Hence, these metrics can be computed locally at the PC that captures the packet trace.

3.4.3 Multiple Packet Traces

In order to compute unidirectional network level performance metrics like delay and loss, it is necessary to correlate the information in multiple packet traces. Thus, information from the packet traces collected at various measurement points must at regular intervals be exported to a central host for post-processing.

To recognize the same packet in several packet traces it is necessary to use a combination of IP and transport protocol header fields like source and destination IP address, source and destination port number, and identification field [Fraleigh]. The delay of a packet from one measurement point to another measurement point is then determined from the associated timestamps. A packet seen at an ingress measurement point that is not observed at its corresponding egress measurement point within a given time-out interval is defined as lost.

3.5 Handling and Post-processing of Packet Traces

This section compares various approaches to post-processing of packet traces in context of the scenario shown in Figure 3.4. A measurement unit capable of capturing packet traces is located at every ingress and egress link of the network domain.

Figure 3.5 shows the various steps in the handling and post-processing of measurement data.

The raw packet trace captured at a measurement point can be grouped and sorted according to various schemes. Examples of ways to group and sort the packets contained in a packet trace include the following:

- No grouping or sorting of packets;
- Group packets according to end-to-end flows [Claffy95];
- Group packets that follow a certain end-to-end path;
- Group packets by incoming and outgoing interface of a node;
- Group packets by the application that generated the packet.

The next step in the post-processing of measurement data is to reduce the volume of the measurement data. Data reduction techniques can be divided into two classes:

- Class 1 – throw away any unwanted information contained in the packet trace. The trace can be filtered such that only selected packets are kept, e.g. packets sent between certain source and destination pairs. Further, any field in the packet header that is not interesting in a given context can be thrown away.

- Class 2 – statistical reduction, e.g. computing the average and variance for selected metrics.

The disadvantage of any data reduction is that single item information in the original measurement data is lost and cannot be reconstructed from the processed measurement data.

The post-processing of measurement data, as shown in Figure 3.5, can be performed locally at the measurement unit (local data reduction) or at a central host (central data reduction). Further, the local host can perform the post-processing "off-line" or "real-time". Thus, the following strategies to process the measurement data must be considered:

Central Processing of Measurement Data
The measurement units collect and store raw packet traces captured during a given measurement period. Then the raw data is exported to the central host where post-processing of the measurement data is performed.

Local "off-line" Processing of Measurement Data
The measurement units collect and store raw packet traces to permanent storage locally. After each measurement period, the raw measurement data is post-processed and only the processed data is exported to the central site.

Local "real-time" Processing of Measurement Data
The measurement units extract relevant information from every packet and perform the post-processing in real-time without writing raw data to permanent storage. Thus, only processed data is stored locally and exported to the central host.

Some selected examples of post-processing of raw packet traces are presented to illustrate the resources required by the various approaches. The following assumptions are made for the examples:

- n measurement units capture packet traces at selected measurement points. Each unit captures traces from two links (e.g. each direction of a bi-directional link) with capacity, c [Mb/s], at full line rate. The links carry traffic with an average packet length, m [Bytes]. Thus, the arrival rate of packets to each unit equals $\lambda_i = \frac{c}{\overline{m} \cdot 8}$. Let $n = 20$, $\overline{m} = 200$ [Bytes] and $c = 155$ [Mb/s].

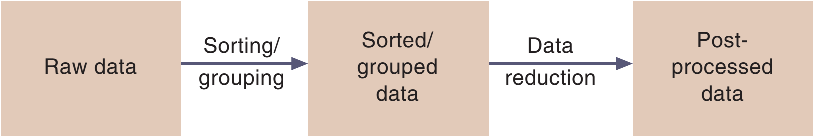


Figure 3.5 Post-processing of measurement data

- The measurement period, t, considered has a duration of 60 minutes.

The following methods for grouping, sorting and data reduction techniques of raw packet traces are considered:

Method A) *Unidirectional performance without data reduction*
All available information about packets, including both endogen and exogen attributes, is needed. Assume that b bytes are required to store all information about a single packet. Let b be equal to 64 bytes.

Method B) *Unidirectional performance – filtering of packets*
All available information about every packet satisfying certain requirements are kept (e.g. type of service equal to a certain value). Assume that ten percent of the packets meet the requirements.

Method C) *Flow records*
All information about every flow is collected. Assume that on the average a flow consists of f packets and that b bytes are required to store all information about a flow. Let f be equal to 40 packets per flow. It may be noted that the number of packets per flow depends on how a flow is defined.

Method D) *Single point metrics*
Statistical data reduction of the measurement data is performed. To determine the average and variance, $\sum l_i$ and $\sum l_i^2$ are computed for certain metrics (e.g. throughput, packet length, interarrival time etc.). Let m denote number of variables computed while k is the number of bytes required to store each of the variables. Let m and k be equal to 100 metrics and ten bytes, respectively.

Table 3.3 shows the data volumes stored locally at each monitor, totally exported and stored centrally by the various methods described above. Note that which approach to apply depend on the performance metrics to be observed.

Table 3.3 Data volumes (locally/exported/centrally) [Gbytes]

Method	Central processing[7]	Local "off line"	Local "real-time"
A	22.3 / 446 /446	22.3 / 446 / 446	22.3 / 446 / 446
B	22.3 / 446 / 44.6	22.3 / 44.6 / 44.6	2.2 / 44.6 / 44.6
C	22.3 / 446 / 11.2	22.3 / 11.2 / 11.2	0.6 / 11.2 / 11.2
D	22.3 / 446 / 0.00002	22.3 / 0.00002 / 0.00002	0.000001 / 0.00002 / 0.00002

Obviously, the actual handling and post-processing of measurement data depends on the ultimate usage of the measurement data and must be planned carefully. Hence, there are many options, and the analyses presented in this section merely illustrate some of the possibilities.

4 A Conceptual Model for IP Measurements

4.1 Introduction

There is an increasing need to define network level performance parameters precisely. For instance service level agreements must state exactly what is being measured. This situation is currently being addressed by the IETF working group IP performance metrics (IPPM) [Paxson96] [RFC2330] and ITU-T [I.380]. This chapter presents a conceptual model for IP measurements [Viken2000] that allows precise definitions of network level performance parameters. The foundation of the model is simple well-defined operations and functions on sets of events. The conceptual model presented is appropriate for packet-switched networks but the nomenclature in this chapter is for IP networks. The model will be used to give precise definitions of previously loosely defined concepts.

4.2 Network Topology

Graph theory is used to describe the topology of a given network domain. The topology of a network domain is modeled by a directed[8] graph $G = (V, E)$ where $V = \{1, 2, ..., n\}$ and $E \subseteq V \times V$ are the set of nodes and directional links, respectively.

V denotes the set of nodes that represent the routers, switches and hosts in the network domain and the outside world. The set of nodes, V, can be divided into four subsets:

- I denotes the set of ingress nodes, $I \subseteq V$, that consists of nodes where packets enter the network domain.

- O denotes the set of egress nodes, $O \subseteq V$, that consists of nodes where packets leave the network domain.

- X denotes the set of external nodes, $X \subseteq V$, that consists of nodes that symbolize the outside world of the network domain.

- M denotes the set of internal nodes, $M \subseteq V$, that consists of nodes that are neither external, ingress nor egress nodes.

Note that a node, $n \in V$, can be both an ingress and an egress node, consequently $I \cap O \neq \Phi$.

E, $E \subseteq V \times V$, represents the set of directional links that connect the nodes. A directed link, $l \in E$, has certain properties like transmission capacity and propagation delay. The propagation delay is mainly determined by the physical distance to the next remote node.

Note that the entire network, a given network domain or any chosen part of a network can be represented by this notation, see example in Figure 4.1 (for simplicity the directed links are not shown but only indicated by arrows). For related work on graph-based models of large networks, see e.g. [Calvert97].

4.3 Paths

A packet that enters the network domain, $G = (V, E)$, follows a certain path, $\pi_{(i,j)k}$, from node $i \in V$ to node $j \in V$. Index k indicates that there are several possible paths from node i to node j. $\pi_{(i,j)k}$ is defined by the sequence of nodes

[7] *Assume that raw packet traces are deleted after the post-processing has been performed.*

[8] *Parallel directed links from node A to node B cannot be represented by this notation. That is, they are represented as one link.*

that are traversed along the path from node i to node j. The links traversed are implicitly defined. If node e is included on the path $\pi_{(i,j)k}$, $e \in \pi_{(i,j)k}$ is true. The set of links traversed along the path $\pi_{(i,j)k}$ is defined by the function

$$\chi(\pi_{(i,j)_k}) = \left\{(n_1, n_2) \in E \mid n_1, n_2 \in \pi_{(i,j)_k}\right\} .$$

The path shown in Figure 4.1 is denoted by $\pi_{(1,7)_1} = (1, 2, 9, 11, 6, 7)$. The set of links on the path is given by $\chi(\pi_{(1,7)_1}) = \{(1, 2), (2, 9), (9, 11), (11, 6), (6, 7)\}$.

The routing algorithm determines the actual path a packet follows through the network domain. The conceptual model will not be used to specify the routing algorithm.

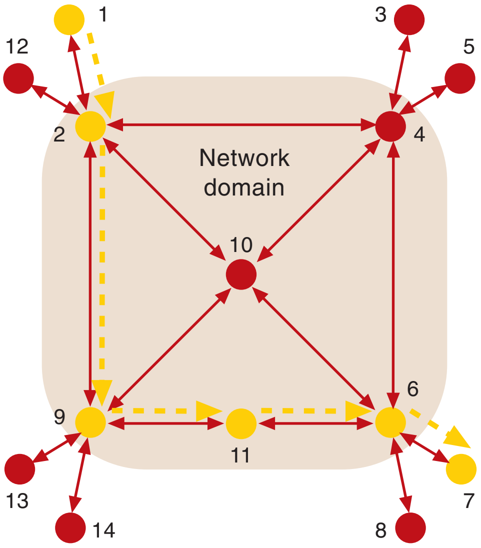


Figure 4.1 An example of a path from node 1 to node 7

4.4 Packets, Events and Sets

Events that occur to packets traversing the network domain, $G = (V, E)$, form the basis for all measurement data that can be collected. Set theory is applied to describe sets of events that occur in the network domain.

First, the fundamental events experienced by packets in a packet-switched network are described. A packet enters the network domain at an ingress node, follows a given path $\pi_{(i,j)k}$ through the network and a successfully[9] carried packet leaves the network domain at an egress node. A packet is delayed through each node $n \in \pi_{(i,j)k}$ and link $l \in \chi(\pi_{(i,j)k})$ along the path. The delay along a given path consists of processing delay, queueing delay, transmission time and propagation delay.

Network congestion causes buffer overflow in nodes and leads to packet loss. Further, nodes may also drop packets intentionally. Packets can be dropped by buffer management and packet scheduling algorithms at intermediate nodes or dropped at the receiver's end if the end-to-end delay is too large. In the conceptual model, packet loss includes both lost and dropped packets.

Packets can also be lost because of transmission errors on the links. The transmission error rate is normally not significant in backbone networks with optical links. On the other hand, radio links can experience a high transmission error rate.

In order to describe the events occurring in the network domain in a precise and concise way, three fundamental set types that form the foundation for creating supersets and subsets of events that satisfy more complex properties, are defined. For a given time interval the events that have occurred at each node are classified in the following three fundamental sets[10]:

- $R_n(t_1, t_2)$ represents the set of packets received by node $n \in V$ in time window $[t_1, t_2>$.
- $S_n(t_1, t_2)$ denotes set of packets sent from node $n \in V$ in time window $[t_1, t_2>$.
- $X_n(t_1, t_2)$ denotes set of packets lost at node $n \in V$ in time window $[t_1, t_2>$.

The fundamental sets of events are illustrated in Figure 4.2.

Note that the sets of events observed by a node can be specialized depending on the internal architecture of the node.

In addition, at a given point in time packets can be in transit between two nodes or already lost due to transmission errors on the link.

[9] *In this context, a successfully carried packet is a packet that was neither lost in a node nor on a link. That is, the content of the packet is not evaluated.*

[10] *Note that the membership of packets in a certain set is a matter of definition.*

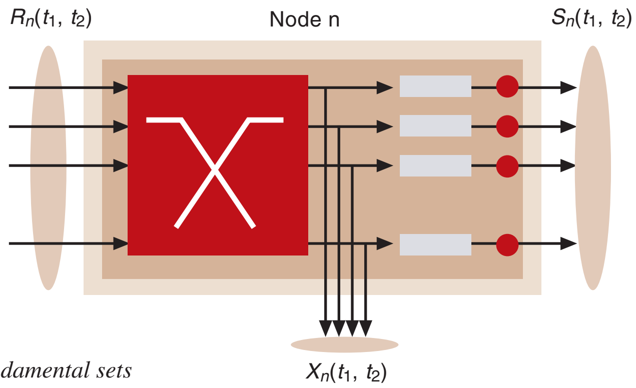


Figure 4.2 Fundamental sets of events as observed at node n

4.5 Packet Attributes and Characterization of Packets

A packet travelling through the network domain is characterised by certain attributes. For a packet[11], *p*, the following functions are examples of definitions that characterize packet properties:

- *src*(*p*) – returns the source IP address of packet *p;*
- *dst*(*p*) – returns the destination IP address of packet *p;*
- *pri*(*p*) – returns the priority level of packet *p;*
- *m*(*p*) – returns the total length of the IP packet *p* in bytes;
- *pth*(*p*) – returns the path that packet *p* shall follow through the network domain. The path is defined by the sequence of nodes that should be traversed by the packet as determined by the routing algorithm;
- *lnk*(*p*) – returns the links that should be traversed along the path of packet *p.*

Generally, every header field of a packet corresponds to an attribute. Note that whether a certain function exists or not depends on the context[12]. Further, functions that return the time a given event took place are defined. These functions are needed to describe network-level metrics that depend on the time certain events happened, as are metrics like unidirectional packet delay and round-trip time.

- $t_a(p, n)$ denotes the time of arrival for packet $p \in R_n(t_1, t_2)$ at node $n \in V$. The time of arrival is defined as the time when the last bit of packet *p* was received by node *n*.
- $t_d(p, n)$ denotes the time of departure for packet $p \in S_n(t_1, t_2)$ from node $n \in V$. The time of departure is defined as the time when the last bit of packet *p* was transmitted from node *n*.

For a packet, *p*, that was lost in node *n* or on link $l = (n, k)$ the time the loss occurred is referred to as $t_a(p, n)$ and $t_d(p, n)$, respectively.

By using the definitions from the previous sections of this chapter, various sets of events that satisfy more complex properties can now be expressed.

4.5.1 Unusual Network Behavior

Unusual network behavior[13] [Paxson97] includes packet re-ordering, packet misdirection, packet replication and packet corruption. These unexpected behaviors can all be expressed by the model. Note that the model does not explicitly define how to handle unusual network behavior. However, the model allows various approaches to be precisely expressed. Below are some examples of how the conceptual model is applied.

- A packet, *p*, is misdirected if it is received by a node, *n*, that is not a part of its path, *pth*(*p*). The set of misdirected packets received by node *n* in time window $[t_1, t_2>$ is defined by $\{p|p \in R_n(t_1, t_2, \wedge n \notin pth(p)\}$.
- A packet, *p*, is replicated if the network delivers multiple copies of the same[14] packet. The set of replicated packets received by node *n* in time window $[t_1, t_2>$ is defined by $\{p|p \in R_n(t_1, t_2) \wedge \exists p_j \in R_n(t_1, t_2)$ such that $p \equiv p_j\}$.

[11] *A packet, p, is considered to be unique. That is, it is assumed that packets are uniquely identified. However, in a real network the ability to uniquely identify a packet depends on which attributes that are available.*

[12] *In a real network, the measurement instrumentation decides which attributes that are observable. However, the conceptual model provides a flexible framework that can be adjusted according to an actual measurement set-up.*

[13] *It may be noted that packet re-ordering is usual behavior in a network with service differentiation. Further, packet corruption not detected from the packet header checksum is not considered.*

[14] *The definition of multiple copies of the same packet depends on which packet attributes that are available. Generally, two packets are equal if all header fields are equal. This is denoted by $p_i \equiv p_j$. Note that packets with equal attributes can be differentiated by the time of observation.*

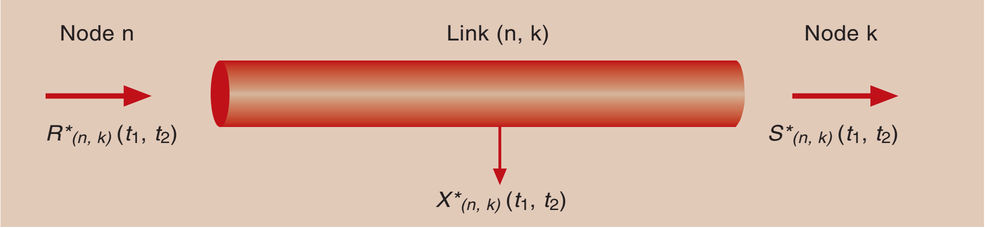


Figure 4.3 Sets of events observed on a link

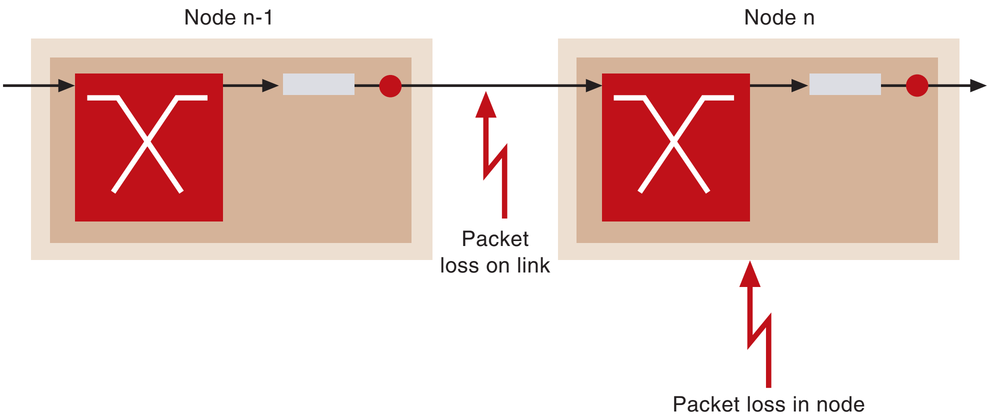


Figure 4.4 Unidirectional packet loss

Table 4.1 Representation of unidirectional packet loss

Description	Definition
$I(p, n)$, indicator variable for loss of packet p in node n.	$I(p,n)=\begin{cases}1 \mid p\in X_n(t_1,t_2)\\ 0 \mid p\notin X_n(t_1,t_2)\end{cases}$ (4.1)
$I^*(p, l)$, indicator variable for loss of packet p on link $l = (n–1, n)$.	$I^*(p,l)=\begin{cases}1 \mid p\in X^*_l(t_1,t_2)\\ 0 \mid p\notin X^*_l(t_1,t_2)\end{cases}$ (4.2)

4.5.2 Sets of Events Observed on a Link

Events that have occurred on a certain link $l = (n, k) \in E$ are described by the following three subsets, see Figure 4.3:

- $R^*_{(n,k)}(t_1, t_2) = \{p | p \in S_n(t_1, t_2) \wedge (n, k) \in \mathrm{lnk}(p)\}$ represents the set of packets sent into the link $l = (n, k) \in E$ from node $n \in V$ in the time window $[t_1, t_2>$.

- $S^*_{(n,k)}(t_1, t_2) = \{p | p \in R_k(t_1, t_2) \wedge (n, k) \in \mathrm{lnk}(p)\}$ denotes the set of packets received from the link $l = (n, k) \in E$ by node $k \in V$ in the time window $[t_1, t_2>$.

- $X^*_{(n,k)}(t_1, t_2) = \{p | p \in S_n(t_1, t_2) \wedge p \notin R_k(t_1, t_2 + \text{prop. delay}) \wedge (n, k) \in \mathrm{lnk}(p)\}$ is the set of packets lost on the link $l = (n, k) \in E$ because of transmission errors in the time window $[t_1, t_2\}$.

Note that packets in transit cannot be observed.

4.6 Unidirectional Network Level Performance

Using the conceptual model, unidirectional network-level performance parameters can be precisely described. This section illustrates the use of the model to describe packet loss.

4.6.1 Unidirectional Packet Loss for Individual Packets

A packet travelling through a network domain can be lost in a node or on a link, as shown in Figure 4.4.

Table 4.1 shows the representation of packet loss for individual packets.

Consider a packet, p, travelling along a path, $pth(p)$, from node i to node j. The packet can be lost at an intermediate node or on a link in the end-to-end path. Thus, the indicator variable of end-to-end loss of packet p, $I(p)$, is the sum of the indicator variables of each intermediate node and link along the path, $pth(p)$.

Table 4.2 Representation of unidirectional packet loss rate

Description	Definition	
$\eta_n(t_1, t_2)$, average loss rate for node n.	$\eta_n(t_1,t_2)=\dfrac{\sum_{p\in R_n(t_1,t_2)} I(p,n)}{t_2-t_1}$	(4.x)
$\eta^*_1(t_1, t_2)$, average loss rate for link $l = (n-1, n)$	$\eta^*_l(t_1,t_2)=\dfrac{\sum_{p\in R^*_{(n-1,n)}(t_1,t_2)} I^*(p,l)}{t_2-t_1}$	(4.x)

Table 4.3 Representation of unidirectional packet loss ratio

Description	Definition	
$Prob_n(loss)$, loss ratio for node n.	$Prob_n(loss)=\dfrac{\sum_{p\in R_n(t_1,t_2)} I(p,n)}{\lvert p\in R_n(t_1,t_2)\rvert}$	(4.6)
$Prob^*_l(loss)$, loss ratio for link $l = (n-1, n)$.	$Prob^*_l(loss)=\dfrac{\sum_{p\in R^*_{(n-1,n)}(t_1,t_2)} I^*(p,l)}{\lvert p\in R^*_{(n-1,n)}(t_1,t_2)\rvert}$	(4.7)

$$I(p)=\sum_{n\in pth(p)} I(p,n)+\sum_{l\in \mathrm{lnk(p)}} I^*(p,l) \qquad (4.3)$$

Hence,

$$I(p)=\begin{cases}1 & \text{packet } p \text{ was lost somewhere on its end-to-end path}\\ 0 & \text{packet } p \text{ was successfully carried end-to-end}\end{cases} \qquad (4.4)$$

4.6.2 Average Loss Rate

Average loss rate is the number of packets that are lost in a given time interval divided by the duration of the time interval. The loss rate can be defined e.g. end-to-end along a path, for an outgoing link in a node or for a certain node.

The average unidirectional packet loss rate is represented as shown in Table 4.2.

The average loss rate for packets travelling from a given source a to a certain destination b is considered. The average end-to-end loss rate is defined for the time window $[t_1, t_2>$ and packets that are still in transit at time t_2 are excluded. Hence, $\{p|p \in R_b(t_1, t_2) \wedge src(p) = a \wedge dst(P) = b\}$.

The average end-to-end loss rate for packets travelling along the path $\pi_{(i,j)_k}$ is defined by:

$$\eta_{a,b}(t_1,t_2)=\frac{\sum_{p\in\tilde{P}} I(p)}{t_2-t_1} \qquad (4.5)$$

4.6.3 Unidirectional Packet Loss Ratio

The loss ratio is defined by the portion of packets that was lost. The loss ratio can e.g. be defined for an end-to-end path, a certain node on the path or a given link.

Then the loss ratio for packets travelling from a given source a to a certain destination b is considered. The end-to-end loss ratio is defined for the time window $[t_1, t_2>$. Hence, $P = \{p|p \in R_b(t_1, t_2) \wedge src(p) = a \wedge dst(P) = b\}$.

$$Prob_{a,b}(loss)=\frac{\sum_{\tilde{P}} I(p)}{|P|} \qquad (4.8)$$

$|P|$ is the cardinality of set P.

5 Concluding Remarks

Active and passive measurements have different pros and cons and are supplementary. Thus, an operational measurement and monitoring platform needs to include both active and passive measurements. Passive measurements are required to e.g. monitor SLAs, perform detailed performance measurements and collect data for surveillance, accounting and pricing purposes.

On the other hand, active measurements are vital to detect network failures and to test network services. An operational measurement platform should be tailor-made for observation of specific parameters. Based on the parameters being observed, data reduction techniques must be considered. The ultimate usage of the measurements decides which methods that are possible. Data reduction is especially crucial for passive measurements (packet traces and flow records) that generate huge amounts of data. Large amounts of measurement data should be transported without any disturbance of the network operation.

As differentiation is introduced in IP networks, the benefits and motivation to collect and analyse measurement data will increase. The service level agreements between various actors must state precisely what is being measured and how the measurements are performed. Since in a differentiated services network the traffic is categorized into flow types or priority classes, the performance within each category must be measured. This can easily be achieved both by active and passive measurements. However, it will be more important to measure the performance of real-time and premium services with strict performance requirements than the best-effort services. Thus, measurements of various granularity may be needed for different service qualities.

References

[Brownlee] Brownlee, N. *NeTraMet.* (2001, August 13) [online] – URL: http://www.auckland.ac.nz/net/NeTraMet/

[Calvert97] Calvert, K L, Doar, M B, Zegura, E W. 1997. Modeling Internet Topology. *IEEE Communication Magazine,* 35 (6), 160–163.

[Careces91] Cáceres, R et al. 1991. Characteristics of Wide-Area TCP/IP Conversations. In: *Proceedings of ACM SIGCOMM'91,* 101–112.

[Cflowd] McRobb, D W. *cflowd design.* (2001, August 13) [online] – URL: http://www.caida.org/tools/mea-surement/cflowd/design/design.html

[Claffy95] Claffy, K C, Braun, H-W, Polyzos, G C. *A Parametrizable methodology for Internet traffic flow profiling.* (2001, August 13) [online] – URL: http://www.caida.org/outreach/papers/pmi.html

[Claffy97] Apisdorf, J et al. 1997. OC3MON: Flexible, Affordable, High-Performance Statistics Collection. In: *Proceedings of INET`97.* (2001, August 13) [online] – URL: http://www.isoc.org/isoc/whatis/confer-ences/inet/97/proceedings/longtoc.HTM. Kuala Lumpur, Malaysia.

[Claffy98] Claffy, K C, Miller, G, Thompson, K. 1998. The nature of the beast: recent traffic measurements from an Internet backbone. In: *Proceedings of INET'98.* (2001, August 13) [online] – URL: http://www.caida.org/outreach/papers/Inet98/

[Coral] *Cooperative Association for Internet Data Analysis (CAIDA), CoralReef.* (2001, August 13) [online] – URL: http://www.caida.org/tools/measurement/coralreef/

[Fraleigh] Fraleigh, C et al. 2001. Design and deployment of a passive monitoring infrastructure. In: *Proceedings of PAM2001,* Amsterdam, The Netherlands.

[Graham] Graham, I D et al. 1998. Nonintrusive and Accurate Measurements of Unidirectional Delay and Delay Variation on the Internet. In: *Proceedings of INET'98.* (2001, August 13) [online] – URL: http://www.comms.uab.es/inet99/inet98/longtoc.htm

[Heegaard] Heegaard, P, Viken B Å. 2001. A distributed test environment for IP performance evaluation. *Telektronikk*, 97 (2/3), 245–268. (This issue.)

[I.380] ITU. 1998. *Internet Protocol Data Communication Service – IP Packet Transfer and Availability Performance Parameters.* Geneva. (Draft ITU-T I.380.)

[Lindh] Lindh, T. 2000. An architecture for embedded monitoring of QoS parameters in IP based virtual private networks. In: *Proceedings of the 15th Nordic Teletraffic Seminar,* Lund, Sweden, 115–124.

[MOAT] *National Laboratory for Applied Network Research, Measurement and Operations Analysis Team (NLANR/MOAT).* (2001, August 13) [online] – URL: http://moat.nlanr.net/

[McGregor] McGregor, A, Braun, H-W, Brown, J. 2000. The NLANR Network Analysis Infrastructure. *IEEE Communications Magazine,* 38 (5), 122–128.

[NAI] *NLANR/MOAT Network Analysis Infrastructure (NAI).* (2001, August 13) [online] – URL: http://moat.nlanr.net/NAI/

[NIMI] Paxson et al. 1998. An Architecture for Large-Scale Internet Measurement. *IEEE Communications,* 36 (8), 48–54.

[NetFlow99] Cisco Systems, Inc. *NetFlow services and Applications. White paper.* (2001, August 13) [online] – URL: http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.pdf

[ocxmon] *NLANR/MOAT Passive Measurement and Analysis.* (2001, August 13) [online] – URL: http://moat.nlanr.net/PMA/

[Pasztor] Pasztor, A, Veitch, D. 2001. A precision infrastructure for active probing. In: *Proceedings of PAM2001,* Amsterdam, The Netherlands.

[Paxson96] Paxson, V. 1996. Towards a Framework for Defining Internet Performance Metrics. In: *Proceedings of INET'96,* Montréal, Canada.

[Paxson97] Paxson, V. 1997. End-to-End Internet Packet Dynamics. In: *Proceedings of SIGCOMM'97,* Cannes, France.

[Prycker95] Prycker, M. 1995. *Asynchronous Transfer Mode: Solution for Broadband ISDN.* 3rd ed., London, Prentice Hall.

[RFC792] IETF. 1981. Postel, J B. *Internet Control Message Protocol.* (Internet RFC 792.)

[RFC1305] IETF. 1992. Mills, D L. *Network Time Protocol (Version 3) Specification, Implementation and Analysis.* (Internet RFC 1305.)

[RFC2330] IETF. 1998. Paxson, V et al. *Framework for IP Provider Metrics.* (Internet RFC 2330.)

[RIPE] Uijterwaal, H et al. 1998. Internet Delay Measurements using Test Traffic: First results. In: *SANE98 Conference.* http://www.ripe.net/ripencc/mem-services/ttm/Talks/9811_sane.ps.gz, Maastricht, Netherlands.

[Surveyor] Kalidindi, S, Zekauskas, M. 1999. Surveyor: An Infrastructure for Internet Performance and Measurements. In: *Proceedings of INET'99.* San Jose. Surveyor Project homepage (2001, August 15) [online] – URL: http://www.advanced.org/surveyor/

[Tcpdump] *tcpdump software available from Lawrence Berkeley Laboratory Network Research Group.* [online] – URL: http://www-nrg.ee.lbl.gov/

[Viken2000] Viken, B Å, Emstad, P J. 2000. A conceptual model for Internet measurements applied to data from OCXmon/Coral monitors. In: *Proceedings of 15th Nordic Teletraffic Seminar,* Lund, Sweden, 43–54.

[Viken99] Viken, B Å. 1999. Passive measurement of Internet traffic from SCi-net'98. In: *Proceedings of the Norwegian Informatics Conference 99 (NIK'99),* Trondheim, 259–270.

A Distributed Test Environment for IP Performance Evaluation

POUL E. HEEGAARD AND BRYNJAR Å. VIKEN

Poul E. Heegaard (37) is Senior Research Scientist at Telenor R&D, Trondheim. His research interests are within the areas of traffic and dependability evaluation of telecommunication systems. He has a special interest in speedup simulation techniques for assessment of systems with rare events, and monitoring and measurements of IP networks. He received his Master's degree in 1989 and his Dr.Ing (PhD) in 1998 in Telematics from the Norwegian University of Science and Technology (NTNU). Heegaard holds an adjunct (20 %) associate professorship in simulation at the Department of Telematics, NTNU. From 1989 to 1999 he was research scientist at SINTEF Telecom and Informatics.

poul.heegaard@telenor.com

Brynjar Å. Viken (31) is Research Scientist at Telenor Research and Development, Trondheim. His research interests are performance measurements and analysis of communication networks with a special interest in IP networks. He is currently pursuing a PhD at the Norwegian University of Science and Technology.

brynjar-age.viken@telenor.com

As input to IP Traffic Engineering it is required to conduct measurement both on live networks and test networks. This paper presents a distributed test platform currently being used for QoS performance testing in an experimental, IP based communication platform that will provide differentiated services. The measurement platform developed for QoS tests consists of two main components; (i) DAG monitors to derive performance measures like end-to-end packet loss and one-way packet delay accurately, and (ii) GenSyn, a Java-based traffic generator running on dedicated PCs. This testbed configuration is a very flexible platform that opens for doing many exciting and controlled QoS performance evaluation measurements in an IP network.

Most testbeds have no, or very low traffic load, so traffic generators are required. The traffic mixture that is generated by GenSyn is controllable, reproducible, synthetic traffic according to an aggregation of stochastic application models. Currently available are models of web and FTP clients that generate TCP traffic by downloading pages and files from actual web servers, and models that generate UDP traffic from a video server (using MPEG), from voice over IP (VoIP), and in a Constant Bit Rate (CBR) stream. Besides generating traffic, there is a need for accurate traffic measurements in order to derive performance measures like unidirectional end-to-end packet loss and delay. This has been achieved by deploying PCs dedicated to traffic monitoring using specialized hardware, so-called DAG PCI cards.

For the purpose of QoS testing of new applications and network mechanisms in IP networks, several test scenarios are defined. The scenarios specify the application mixture (VoIP, VoD, FTP, web, TV, etc.) and/or protocol mixture (TCP, UDP), load level, traffic matrix, and network configuration (DiffServ, Best Effort etc.).

The experiences from running experiments in such a distributed test environment revealed a need for a support system assisting the analysts in setting up the experiments, collecting data, and post processing it. Such support functions are under development and are briefly presented in this paper.

1 Introduction

A test platform is required for Quality of Service testing in IP based networks that carry all types of services. The services that are provided can e.g. include Internet access, home office, telephony, video and TV distribution. These services have different traffic characteristics and Quality of Service requirements and the IP network will therefore require some support for differentiated QoS provisioning. Such a test platform should include generation of realistic traffic and monitor functions that are able to study the details of traffic streams in the network. This paper presents a flexible test platform currently being used for QoS performance testing in an experimental, IP based, communication platform that will provide differentiated services.

For the purpose of QoS testing of new applications and network mechanisms in IP networks, a generator of controllable, re-producible, scalable, synthetic but realistic traffic is required. This is the motivation for the development of GenSyn – a generator of synthetic IP traffic implemented in Java. The generator will typically produce traffic in a controlled testbed environment where there are few real users and a corresponding low traffic load. Realistic, controllable, and reproducible background traffic load is of great importance in order to test the QoS of new applications, i.e. interactive video. In addition, to increase the understanding and get experience with the configuration of new QoS network mechanisms the network must be offered a high load of heterogeneous traffic mixture.

Besides generating traffic, a set of performance parameters must be monitored to study the service performance defined at IP layer or transport layer (UDP/TCP), or even higher layers, e.g. for real-time services the parameters can be IP packet delay, jitter and loss, while the quality of TCP connections can be measured as throughput at TCP or IP layers. Other performance characteristics can be derived based on these parameters, e.g. the service availability can be defined as the state where the end-to-end delay is less than a specified limit, and the measurement accumulates the fraction of the time the system is in this state. A flexible measurement set-up has been achieved by deploying PCs dedicated to passively monitor traffic using specialized hardware, so-called DAG PCI cards. However, the specification of a measurement and monitoring platform depends on what performance parameters that are to be observed.

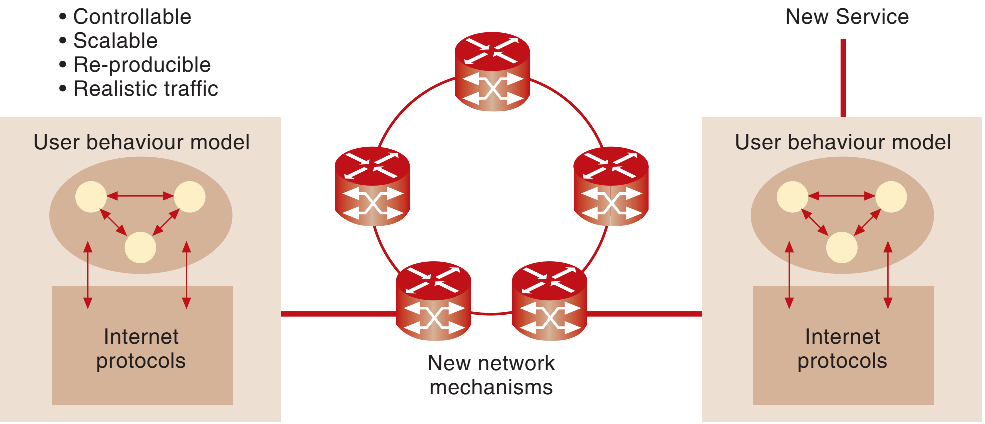


Figure 1 Load generation in an IP test environment

In this paper, the GenSyn measurement platform is described. In Section 2 details about the GenSyn traffic generator are given. The general modelling framework of GenSyn is described in Section 2.1, while Section 2.2 contains the current available interface modules and examples of source models; including FTP client, VoIP and a combined user model. In Section 2.3 the implementation details of GenSyn are given and performance constraints are discussed. Section 3 describes details of a distributed test platform and different test scenarios, including examples of results. Section 4 describes support systems for conducting distributed experiments, GenSyn Designer for setting up an experiment, and GenSyn DataReporter for post-processing of measurement data. Finally, the paper closes with some general remarks and a list of ongoing and further work in Section 5.

2 GenSyn – a Java-based Traffic Generator

2.1 The Modelling Framework

Different source modelling approaches can be considered to describe a typical Internet source:

- *Trace* – replay of recorded stream of IP packets, i.e. a stream of IP packets obtained from measurements is replayed and offered to the test network (e.g. replay of a tcpdump-file). If the recorded stream contains traffic from elastic sources (e.g. TCP connections) the replay will not be representative unless the congestion situation in the network is exactly the same. This will rarely be the case in a network with a mixture of traffic streams.

- *Aggregate generator* (“black box”) – generation of IP packets according to a parametric stochastic process. If a recorded aggregate of packets from elastic sources is used to determine the parameters of the model, the same problem as described in Trace above will still be present.

- *User behaviour model* (“white box”) – generation of IP packets from physically based source models. More detailed measurements than for the two other approaches are required. However, the parameters in the model reflect the user behaviour and hence it is straightforward to change the model if e.g. the number of sources is changed.

For generation of traffic in IP based testbeds both user behaviour models, see e.g. [MTW99], [ChLi99], [Vic98], [Ake99], [BaCr98], and bulk-transfers, e.g. tTCP [MTW99], are being used. GenSyn combines the user behaviour approach with bulk-transfer of data via communication streams. Application of stochastic user behaviour models described by state diagrams introduces flexibility, scalability, and physically interpretable model parameters. This part of the modelling framework is similar to ideas used in a former ATM traffic generator, called ATM100 or STG [HMM93]. However, instead of developing a specialized hardware instrument, GenSyn applies modern Web- and Java-technology and exploits the Internet protocol suite (TCP/IP) that is already available. This means that the generator is only a software process that imitates the user behaviour and dynamically controls the creation and deletion of one or more links (threads) to physical HTTP- and TCP-connections. The generator is not only a simulator; it generates real IP packets that flow through a real (test) IP network.

This section describes the fundaments of GenSyn and the flexible modelling framework with a few examples of use.

2.1.1 Multi-level Stochastic Behaviour

The models described using the GenSyn framework will attempt to reproduce the inner workings of the physical source. This includes many stochastic processes, both human, environmental, and communication equipment. An example of the variety of activity levels in a source is

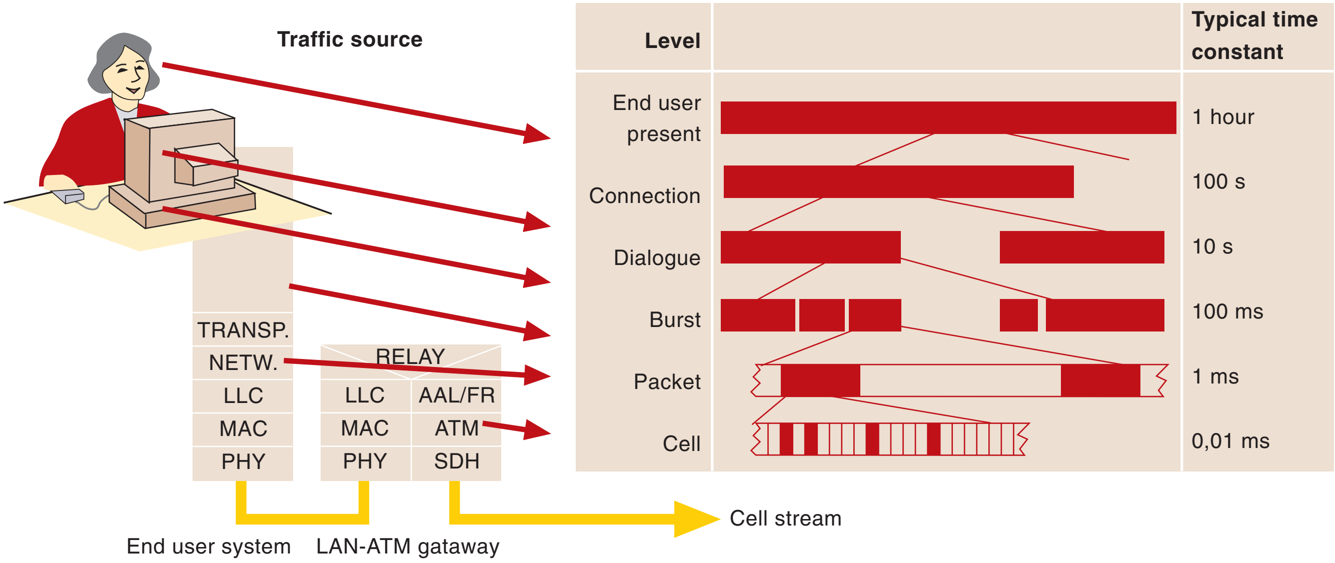


Figure 2 Illustration of different activity levels in a source [Helv95]

illustrated in Figure 2. The example is originally from ATM [Helv95], but this general multi-level behaviour is independent of the communication technology. Consider for instance a telephony user. The user is present (at the office) {end-user present level}, he is making a phone call {connection level}, during the call he is speaking and listening {dialogue level}, when he is speaking he sends voice and takes short breaks {burst level}, when voice is sent it is wrapped in packets {packet level}, each packet is segmented into cells {cell level}. The last two levels are technology dependent. The typical time constants involved on various levels are indicated in the table in Figure 2. Hence, the aggregated packet or cell stream that can be observed on the transport or cell level will have a communication pattern that is generated as a result of many interacting stochastic processes with different time constants. In [HeHo95] and [WTSW97] this multilevel superposition of stochastic processes with different time constants is considered to be an explanation of the observed self-similar behaviour observed in aggregated traffic streams (packet or cell level) on the Internet [LTWW94].

The aggregated stream will be a heterogeneous mixture of traffic from various sources where each source will be influenced by user behaviour, equipment and protocols, see Figure 2.

User behaviour:

- *The end user behaviour* – set-up/disconnect a session, application mixture (web browsing, software downloads, chat, games, email, streaming (video, audio));
- *User-network interaction* – slow variation, interest/impatience, takes a break and returns later due to congestion, cost, etc.;
- *Variation in information stream* – e.g. variable video coding (MPEG).

Equipment and protocols:

- *End user equipment constraints* – access capacity, processor capacity, disk, video coding processing;
- *Communication system constraints* – buffer space, transmission capacity, router capacity;
- *Network mechanisms* – routing strategies, priorities (DiffServ), weighted fair queuing, resource reservations (RSVP, IntServ);
- *Protocols* – e.g. TCP congestion control and avoidance.

2.1.2 Linking Stochastic User Behaviour Models to Real Communication Streams

GenSyn models the user behaviour in a state based source model, while the communication systems are not modelled. The equipment constraints and protocol behaviour are automatically included through the linking of the stochastic processes to the built-in protocol stack on the workstation. This means that no incorrect assumptions about the protocols or network mechanisms will be made, it is for instance not necessary to know the details about the MPEG coding or the TCP slow start mechanisms.

In general it can be said that the modelling framework combines the better of two worlds, it gets the flexibility and scalability of state diagram description with composition of users combined with the accuracy of protocol and network behaviour by using the actual protocol instead of a model.

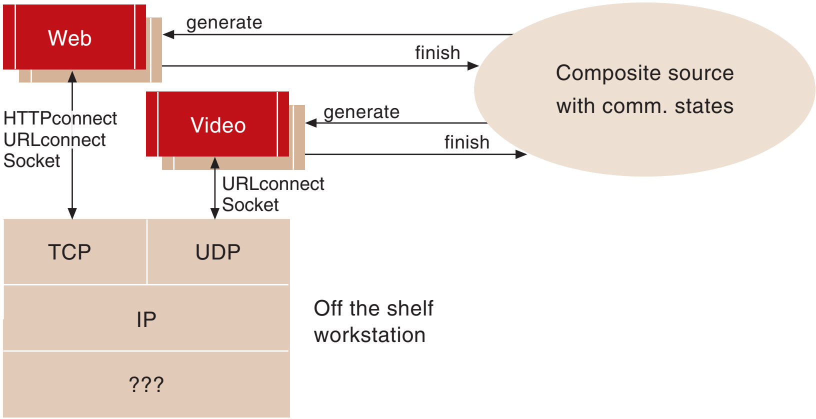


Figure 3 The interface modules are linking the stochastic behaviour to built-in protocol stack

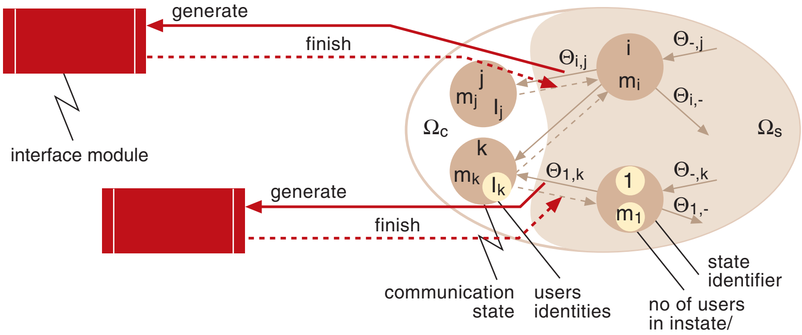
2.1.2.1 Division of State Space

Figure 3 shows a principal sketch of the modelling framework to illustrate the linking of a composite state space description and the protocol stack. This efficient combination is accomplished by dividing the modelling state space into:

- *Stochastic state space*, Ω_S, where the stochastic user behaviour is described by a state machine where each state can contain a collection (composition) of many users[1]; and
- *Communication state space*, Ω_C, which is a "placeholder" for the users that are waiting for response from the communication system.

All transitions from the stochastic to the communication state space, $\Omega_S \rightarrow \Omega_C$, will instantiate an interface module that creates a communication stream to and/or from the workstation. A communication stream is either a TCP connection or a stream of UDP datagrams. The user that instantiated the communication stream will be placed in a communication state and stay there until the communication is completed. Hence, a transition from the communication state space back to the stochastic state space, $\Omega_C \rightarrow \Omega_S$, is initiated on completion of communication and the corresponding pointer to the interface module is removed.

Figure 4 The modelling framework: division of state space



The number of states will not change during the evolution of the generator process; only the number of users in each state will change. All states need an attribute for the current number of users in each state, and the communication states need an additional attribute for the storage of information about the user (or process) identities that await a communication stream to complete. Hence, only a modest increase in the generator process is observed as the number of users increases. This solution is chosen to optimise the scalability against the accuracy in the protocol modelling.

2.1.2.2 Notation

The following notation will be used:

- Ω_S – stochastic state space
- Ω_C – communication state space
- Ω – global state space, $\Omega_S \cup \Omega_C$
- $\mathbf{m}$ – state vector, $\mathbf{m} = \{m_i\}_{i \in \Omega}$
- m_i – number of users in state i
- $\mathbf{I}_i$ – identity vector of users with an open communication stream in state $i \in \Omega_S$
- $\theta_{i,j}$ – state transition rate from state i to state j, $i \in \Omega_S$
- θ_i – total transition rate in state i, $i \in \Omega_S$, $\theta_i = \Sigma_{j \in \Omega} \theta_{i,j}$
- $E(T_i)$ – expected sojourn time $E(T_i) = 1 / (\theta_i m_i)$, $i \in \Omega_S$

In Figure 4, the notation is shown in an example demonstrating the division of state space and the links to the interface modules.

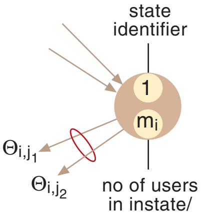
2.1.2.3 Stochastic State Model – the Behaviour of a Single Source

A finite state continuous time Markov process describes the user behaviour. A general state has the following attributes:

- a state identifier, i;
- number of users m_i in state i;

[1] *The state model has to be (semi-)Markovian, i.e. each state sojourn time must be negative exponentially distributed, in order to make the composition of users in each state.*

- a state sojourn time distribution (negative exponential distribution);
- list of transition rates, $\theta_{i,j}$, and probabilities, $p_{i,j} = \theta_{i,j} / \theta_i$;
- list of neighbour states, i.e. states that can be reached in one transition from state *i*.



A state model is semi-Markovian if all states have state sojourn times that are negative exponential distributed, or if neither of the states in the model have more than one non-exponential sojourn time. In case of a state with non-exponential time distribution, an approximation by a phase-type distribution is feasible by substituting this state with a combination of states that have negative exponential time distributions. This means it is possible to model state sojourn times that follow a hyper-exponential, hypo-exponential, or Coxian distribution. All these distributions are a combination of states with negative exponentially distributed state sojourn times.

When all state sojourn times are exponential, the procedure described in Algorithm 1 can be applied to determine the next stochastic event. Observe, however, if a communication stream is completed while waiting for the next scheduled stochastic event to occur, an immediate state change will occur caused by closing the communication stream, see Algorithm 2 for further details.

2.1.2.4 *Communication State Model – Link to the Network*

The communication states are the "placeholders" for all users that currently have an open communication stream. In the case where TCP is used for transmission of packets (e.g. using the FTP or web model), the states sojourn times, T_i, $i \in \Omega_C$, for all users in the communication states are fully determined by the behaviour of the underlying communication system, i.e. the performance of the end user equipment, protocols, network mechanisms. Using UDP for transmission, the state sojourn times are stochastically determined by the distribution included in the corresponding interface module (e.g. CBR, VoIP, or mpeg). However, in both TCP and UDP cases a user will be "locked" in the communication state as long as the communication stream is open, and immediately be removed when the stream is closed. Hence, for user *x* the transition from state *i* in Ω_C to state *j* in Ω_S is considered to be a conditional transition, i.e.

$$\theta_{i,j}(x) = \begin{cases} \infty & \text{comm.stream opened by } x \text{ is closed} \\ 0 & \text{comm.stream opened by } x \text{ is open} \end{cases} \qquad (1)$$

where state $i \in \Omega_C$ and state $j \in \Omega_S$.

Algorithm 1: Update the state vector when the next event is a stochastic event in W_S

(i) Sample the time *T* to next event in W_S, the expected value is

$$E(T) = 1 / \left(\sum_{j \in \Omega_S} \theta_j m_j \right)$$

(ii) Wait *T*

(iii) Sample which state $i \in W_S$ where the next event took place, the probability is

$$\theta_i m_i / \left(\sum_{j \in \Omega_S} \theta_j m_j \right)$$

(iv) Sample the next state *j* from state *i*, the probability is $p_{i,j} = q_{i,j} / q_i$

(v) Move a user from state *i* to state *j* by updating the $m_i - 1$ and $m_j + 1$.

In the communication states a relation, e.g. a process identity, to all opened communication streams must exist. When a communication stream is opened and a new user enters state *i*, $i \in \Omega_C$, the process identity of the interface module related to this state transition needs to be stored. For this purpose an identity vector $\mathbf{I}_i$ is added as a new attribute to the communication states in addition to the list in Section 2.2.3. When a communication stream is closed, e.g. when a file is downloaded, an instantaneous state transition will occur, and a user leaves this communication state ($m_i - 1$). Observe that the number of users in state *i* equals the number of elements in the identity vector in state *i*, $m_i = |\mathbf{I}_i|$. The identity vector, $\mathbf{I}_i$, is implemented as a list of pointers (process identities) to open communication streams.

In Algorithm 2 the addition to Algorithm 1 is given to handle both stochastic events and events triggered upon completion of a communication stream.

The state sojourn time in a communication state may depend on the current congestion situation in the underlying network. In the case of downloading web pages, the congestion, the size and location of the requested page will contribute to the sojourn time. Furthermore, for web- and FTP-downloads an *impatience factor* is defined

Algorithm 2: Update the state vector when the next event is an event in W_C

(i) Sample the time T to next event in W_S as described in step (i) of Algorithm 1.

(ii) If $(\exists k \in W_C) \wedge (T_k < T)$, i.e. a communication stream is closed before the time T elapses, then replace step (iii) of Algorithm 1 with the following:

(iii) The next event takes place in state k in W_C where k is the first stream closed, i.e.

$$k = \left\{ i \middle| i \in \Omega_C \wedge \left(T_i = \min_{j \in \Omega_C} \left(T_j \right) \right) \right\}$$

Step (iv) and (v) are the same steps as in Algorithm 1 except that the $p_{i,j}$ is given as parameters because they cannot be derived from the conditional rate $q_{i,j}(x)$ which is now either ∞ or 0.

for each communication state. This enables the definition of an upper limit on the time spent in the communication state. The impatience factor is formulated as a condition of $\theta_{i,j}(x)$ in (1), and its randomness is defined in the stochastic part of the source model.

2.2 Model Template Examples

This section demonstrates the applicability and the flexibility of the modelling framework by describing the model templates that have been developed using the GenSyn framework. First the interface modules currently available are described, followed by two examples of model templates that imitate web and VoIP user behaviours and a model that combines these. The interface modules handle the actual communication through the IP network.

The framework of GenSyn is not limited to these models, and can easily describe other state oriented models, or change the model parameters. Finally, Section 2.2.3 includes an example of a packet trace collected at a single point in a test network viewing packets generated from several GenSyn processes running a mixture of FTP and voice models.

2.2.1 Available Interface Modules

To make it easy to build new models and create realistic traffic mixtures a set of different interface modules is developed – including e.g. a module for downloading and reading web pages, a module for sending a deterministic stream of UDP, and a module for sending a stream of UDP packets determined by the content of a trace file, e.g. an mpeg coded video stream. This section describes some of the details in these modules.

2.2.1.1 Web Module – a Web Browser

The web module is a simplistic web-browser that downloads web pages from actual web servers. The browser downloads the source file of the web page, parses this, and downloads the embedded objects (e.g. images and java applets) identified on this page. These are also downloaded, in parallel with each other and in parallel with the download of the web source file.

The url addresses are randomly chosen from a list of 2500 addresses from all over the world. This list can easily be changed. For example, as an option, GenSyn offers to dynamically check and update the url list as the generator is running and new pages are visited. This is done by inclusion of some, or all, of the href addresses found when parsing through the source file of a web page.

The download time per web page is constrained by the transport protocol and the network and server conditions. In addition, GenSyn allows the user to set a time-out parameter, *impatience time*, that sets the maximum limit of the download time.

2.2.1.2 Ftp Module – Download a File

The FTP interface module uses http to download a file. The file can be downloaded from any machine that is set up as a web server and read directly into the memory on the receiving machine, i.e. the machine that is hosting the GenSyn process running the FTP client. In contrast to the web interface module in Section 2.2.1.1, see also [Heeg00], the FTP module adds very little to the download time due to processing in GenSyn, and hence the download time is constrained by the server responses and transfer delays. Similar to the web module, the *impatience time* is also a parameter in the FTP module. See Section 4.2 for discussion of GenSyn constraints.

The pointer to the files that can be downloaded by the FTP interface module are given as url addresses in a separate parameter list as input. This list is in a text file that can easily be changed and hence the user of GenSyn can create any empirical file distribution. As an example, the FTP client in the experiment in this paper randomly selects its files from the file size distribution (i.e. *not* the IP packet distribution) depicted in Figure 5 with an average size of 154 kbytes (adopted from [Dan92]).

2.2.1.3 Mpeg Module – Sending UDP Packets According to Trace Data

The mpeg module was developed to have a module that can read a stream of numbers representing a variable bitrate coded video trace, convert to udp packets and send it to a given address. In general this module can take any file that con-

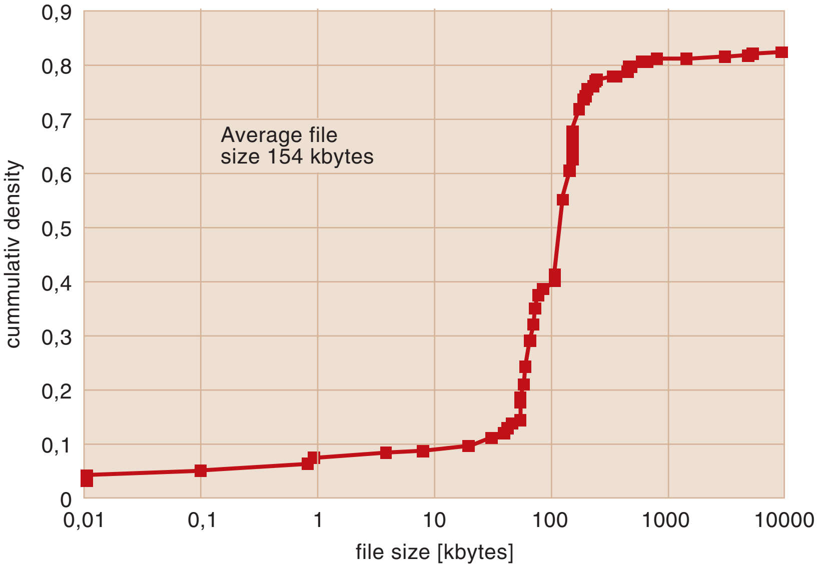


Figure 5 An example of a file size distribution used by the FTP interface module

tains a stream of numbers as an input. The numbers are interpreted as the number of bits that are to be put into a datagram (udp packet) the next period. The module takes as parameters the (fixed) maximum size of the datagram and the (fixed) duration of each period. If the number of bits in the trace file for one period does not fit into one packet, several packets are created and sent back-to-back.

As an example of use, let the trace file be the streams of MPEG-1 coded video sequences made available by Oliver Rose [Rose95]. Each video trace contains n video frames $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ ($n = 40000$). A video frame will then be the sending period. A frame consists of a variable number of bits using MPEG coding and the most common *Group of Picture* pattern *IPPBPPBPPBPPB*, see Figure 6.

The X_i bits in video frame i are converted to Y_i UDP packets of N_P bytes, i.e.$Y_i = \lceil X_i/(N_p \cdot 8) \rceil$. A new video frame is sent every T_P and the size of the video frame is given by the current position in the MPEG trace.

All Y_i UDP packets in position i are sent back-to-back at maximum line speed. A video is randomly, and uniformly, selected among the $K = 19$ different video streams that are available. The packet size N_P and interference time T_F are parameters that can be set by the scenario designer. Default values are set to $N_P = 1024$ bytes and $T_F = 40$ ms.

Although this module originally uses a set of MPEG-1 coded video sequences it is very simple to e.g. use MPEG-2 traces instead, as long as the resulting input trace files have the same format as **X**. In fact, any trace with the format of **X** applies. It is also easy to change the size of UDP packets, and the time between each video frame. This is very convenient when investigating the sensitivity and importance of e.g. video coding techniques and frame segmentation.

2.2.1.4 VoIP and CBR Modules – Sending Constant Stream of Fixed Sized Packets

Both the VoIP and the CBR (from the term constant bit rate) interface modules send a deterministic stream of packets, i.e. a stream of equally sized packets with a constant inter-packet arrival time. Hence, the packet pattern is characterized by the packet size N_p (not included overhead like 8 byte UDP header and 20 byte IP header) and the (constant) time between packets, T_p. This gives a bitrate of (excluded overhead) $8N_p / T_p$. Figure 7 shows an example of a packet pattern generating a 64 kbit/s stream, or 8 packets per second (pps).

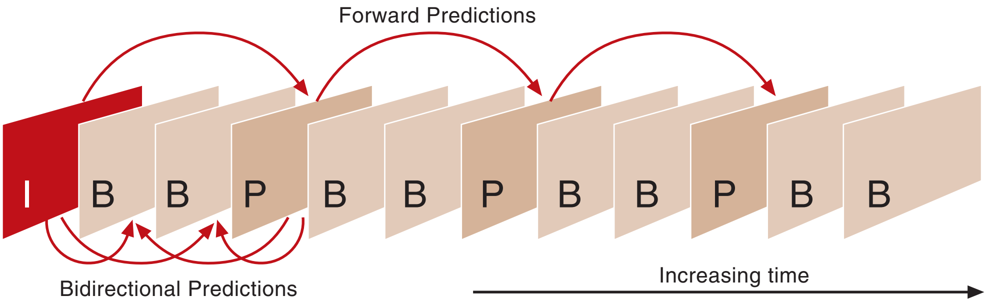


Figure 6 MPEG coded video: An example of a Group of Picture pattern

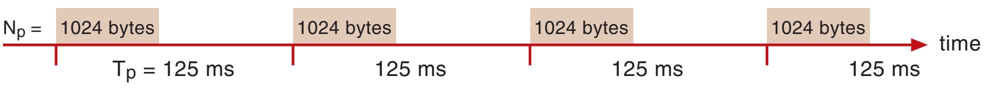


Figure 7 An example of a 64 kbit/s streaming of udp datagrams for the constant packet source

Module	Protocol	Source IP	Source port	Destination IP	Destination port	Packet size	Packet interarrival	Send period
Mpeg	Udp	*Fixed*, given by the machine	*Random*, uniform in (25000, 30000]	*Random* from IP address list	*Parameter* from parameter file	*Frame size is parameter* from parameter file	*Frame interarrival is parameter* from parameter file	*Random*, neg. exp. distributed
Cbr	Udp	*Fixed*, given by the machine	*Random*, uniform in (30000, 35000]	*Random* from IP address list	*Parameter* from parameter file	*Parameter* from parameter file	*Parameter* from parameter file	*Random*, neg. exp. distributed
VoIP	Udp	*Fixed*, given by the machine	*Random*, uniform in (20000, 25000]	*Random* from IP address list	*Parameter* from parameter file	*Parameter* from parameter file	*Parameter* from parameter file	*Random*, neg. exp. distributed
Web	http-tcp	*Fixed*, given by the machine	*Random*, given by http	*Random* from url list	*Fixed*, 80, 90	*Parameter*, file size is given by web page size in url list	*Variable*, given by tcp/ip behaviour	*Variable*, constrained by the network and servers
FTP	http-tcp	*Fixed*, given by the machine	*Random*, given by http	*Random* from file list	*Fixed*, 80, 90	*Parameter*, file size is given in file list	*Variable*, given by tcp/ip behaviour	*Variable*, constrained by the network and servers

Table 1 Characteristics of the current available set of interface modules

The VoIP is a specialisation of the CBR module where the stream of packets has random breaks. Both the time between breaks and the break duration follow a negative exponential distribution. The VoIP module was created to model silence suppression in an audio stream as introduced and described in [Brad69].

2.2.1.5 Overview of Interface Modules

In Table 1 is given an overview of the current set of interface modules. New modules will be developed as requested by new models in new traffic scenarios.

2.2.2 Example of a User Behaviour Model

In Section 2.1 the modelling framework of GenSyn was described. The two-layered modelling consists of a flexible and scalable state diagram approach for describing the user-behaviour and the interface modules described in the previous sections. This means that on top of the interface modules that were introduced in the previous section many source models can be described. In [Heeg01] descriptions can be found of an FTP client that uses the FTP module and a voice model that uses either the VoIP (for modelling of silence suppression) or the CBR module. In [Heeg00] models of a web and video server are given.

In this section a model that combines the FTP and voice models from [Heeg01] will be briefly described to demonstrate another aspect of the flexibility of the GenSyn framework, namely the ability to use more than one interface module in the same model. First the descriptions of the FTP and voice models from [Heeg01] are repeated before the complete combined model is given.

2.2.2.1 The FTP Client

The overall model of the FTP clients describes the user behaviour within and between sessions by the 3-state model in Figure 8. A session is essentially the same as the web sessions defined in [Vic98] as a sequence of packets with less than 30 minutes between two consecutive file requests. The following states are defined:

- *Idle* – the user is in-between sessions.
- *Read* – the user reads the downloaded web page or a file and considers what to do next, download another one, or close the session?
- *Download* – the user opens a connection to a url address, randomly selected from a list of addresses, and downloads this page or file. In the web interface module the page is parsed and all corresponding image files and applets are downloaded.

The *Idle* and *Read* states are stochastic states. The state sojourn times of these two states are sampled from a probability density distribution. Each user that starts to download a file enters the *Download* communication state. A pointer to the interface module that handles the communication stream is added to the identity vector of Download, $\mathbf{I}_{\text{download}}$. When the file, and all its content (text, images, applets), is downloaded, the interface module closes the connection and removes itself from $\mathbf{I}_{\text{download}}$ and the user returns to the *Read* state. The modelling framework enables the (random) setting of an upper limit of the download time, the impatience factor. This factor allows the connection to be closed before the entire web page is downloaded.

The parameters of the state model are extracted from the work described in [Vic98] where an aggregated stream of HTTP connections was separated and broken into web sessions.

- *Time sequence of file downloads* – The Idle state uses the web session separator criterion as the expected sojourn time T_{Idle} ~ neg.exp. (γ); $1/\gamma = 30 \cdot 60 = 1800$ [sec.].

- *Time between requests* – Within a web session the mean time between requests is $\bar{X}$ 42.8 [sec.] with a coefficient of variation equal to $S/\bar{X} = 2.9$. This means that the Read state in the overall model is not negative exponential distributed (in that case the $S/\bar{X} = 1$). This state is therefore substituted by a hyper-exponential distribution with 4 branches, each with different time constants. The parameters are determined to fit the truncated-Pareto model used in [Vic98].

Substituting the *Read* state with four sub-states to model the hyper exponential distribution, the complete model and the model parameters are given in Figure 8.

2.2.2.2 The Voice Model

The voice model that is described does not include control traffic, i.e. call set-up and disconnection signalling. The VoIP model uses a simplified model of a telephony user. The media stream is uni-directional, i.e. the A and B parties are included in the same model but no synchronisation (two-way communication) between the two parties exists.

The user behaviour model consists of two states as illustrated in Figure 9:

- *Idle:* A source generates three calls per hour and each call is generated according to a Poisson process;

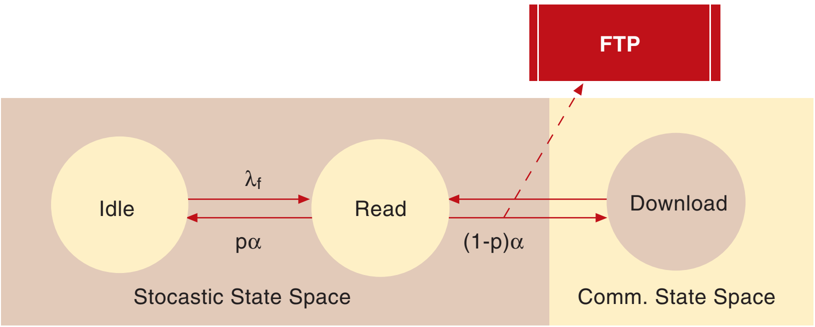


Figure 8 The overall model of a web or FTP client

- *Connect:* The call duration time follows a negative exponential distribution with average of three minutes.

2.2.2.3 The Complete and Combined Model

The two models can be combined on one machine by running either

- *two processes,* one with FTP and one with voice; or

- *one process* where the user behaviour state description is combined into one model.

In the current case, the simplest, and probably the most efficient approach is to run two processes. Even so, this section will look at a com-

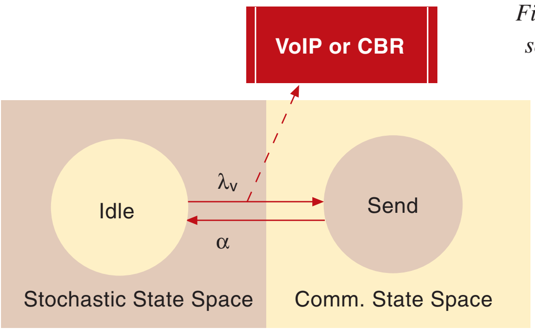


Figure 9 The model of a voice source without call signalling

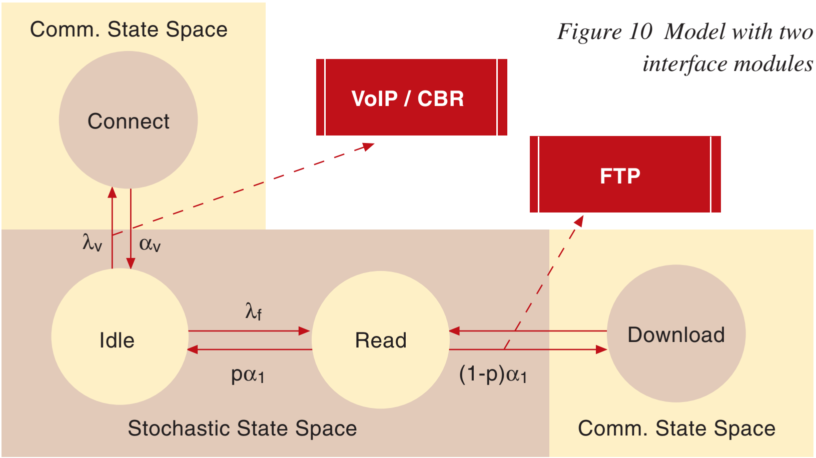


Figure 10 Model with two interface modules

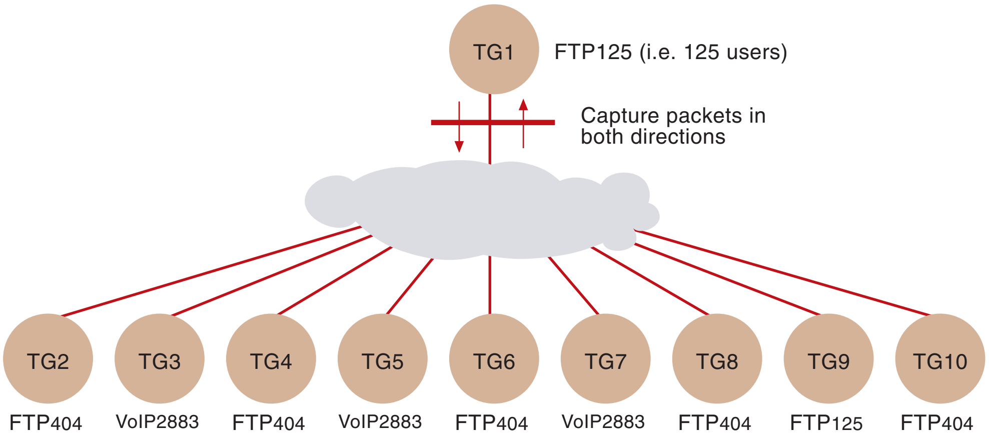


Figure 11 Measurement set-up and GenSyn model deployment

bination to demonstrate that a model can use more than one interface module.

To combine the two models, a common state needs to be identified. In this example the Idle state is an obvious choice. As an implicit assumption a *single* user instance in this combined model cannot be both downloading a file and sending an audio stream at the same time, but the model can contain many users so the process may have many FTP and voice connections open at the same time. An illustration of the combined model is given in Figure 10. Out of the Idle state, the transitions that were described for the separate models are unchanged. However, observe that the transition rates have to be adjusted to provide the requested mixture of TCP and UDP traffic.

2.2.3 Example of a Packet Trace from GenSyn

To demonstrate the use of GenSyn, a packet trace is captured on a single point in a test network. The test network consists of four edge routers connected in a ring topology. To these 4 edge routers there are ten PCs connected that are running an instance of the GenSyn traffic generator with different models. A measurement process is invoked on the interface between GenSyn machine *TG1* and *TGx*, $x = 2, ..., 10$, see Figure 11. All packets in both directions over this interface are collected. The traffic generator, *TG1*, is hosting a GenSyn process running the FTP model, and hence *TG1* sends requests and receives files from *TGx*, $x = 2, ..., 10$ that acts as file servers for *TG1*. At the same time, *TG1* is the file server for all machines *TGx*, $x = 2,..., 10$ that are running an FTP client process. In byte volume, the traffic mixture consists of 85 % TCP and 15 % UDP. A principal sketch of the measurement set-up is given in Figure 11. More details of the measurement platform are given in Section 3.

All packets transmitted and received over the measurement interface for a period of 1800 seconds are collected by tcpdump. The packet trace is post-processed and divided into bins of size 10 milliseconds where the number of bits in each bin is calculated. This time series is denoted $X^{(1)}$. Furthermore, the average traffic over blocksize m bins forms the time series $X^{(m)}$. The variance is calculated for various block sizes, $Var(X^{(m)})$. In Figure 12 the normalized variance, $Var(X^{(m)}) / Var(X^{(1)})$, is plotted for different block sizes. The TCP and UDP traffic is split into two curves and compared to the normalized variance of Poisson process as a reference. The plot shows slowly decaying variance for both UDP and TCP.

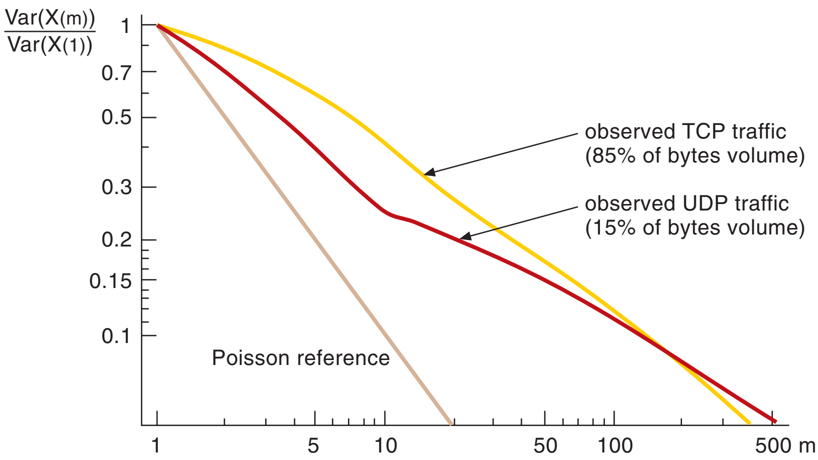


Figure 12 Normalized variance for average traffic over block length

It is important to emphasize that this experiment is included for the purpose of demonstrating the features of GenSyn only, and not to give a full verification of the two model examples. In this section the focus is the modelling framework of GenSyn and characteristics and features of the model examples are out of scope. Building good models and verifying them is an important task and should be treated much more thoroughly in an additional study. However, GenSyn provides a flexible means for modelling and allows the user to develop new models that have the characteristics of interest. The FTP and VoIP models are only two examples of what can be described in the GenSyn modelling framework, see e.g. [Heeg00] for additional examples.

2.3 Implementation and Constraints

The design of GenSyn had the following overall requirements [HeLu99]:

- *Portable* – run on Windows, Linux, Unix, and produce the same results.
- *Distributed* – run in parallel on several workstations – and be easy to distribute.
- *Scalable* – run many active users in parallel on a single workstation.

The first two requirements made Java an attractive choice as the implementation language. Java provides simple, high level, and well-defined interfaces and methods for communication (via APIs) with the underlying protocol stack. This makes it fairly easy to create HTTP and TCP connections and to send streams of UDP packets.

The flip side of the coin is that Java limits the scalability of GenSyn due to two constraints:

- *Time scheduling and granularity.* The sleep function in Java is inaccurate for time granularity in milliseconds, and returns different results on various platforms. This will limit the number of users that can be defined in the model because many users means short time between events in a composite model. The experience so far, running between 300 and 3000 web users on a single processor machine, indicates that the generator running on one workstation should limit the number of users to keep the expected time between stochastic events of at least 10 ms.
- *Threads and memory.* The number of parallel threads that can be run on one single workstation is limited by the available memory. Hence, the GenSyn should be executed on a dedicated workstation. This was expected to be the critical restriction because it introduces an upper limit to the number of simultaneous communication streams, and hence the number of users in the model.

The following section presents a few results of the performance of a single GenSyn process, using a scenario generating both TCP and UDP traffic by running a single instance on a dedicated machine. Knowledge of these constraints are important for instance while specifying a test scenario.

2.3.1 The GenSyn Performance Constraints

2.3.1.1 TCP Traffic – the Throughput Constraints

The web interface module described in [Heeg00] downloads and parses through web pages. This puts a rather heavy burden on the processor and will limit the scaling of a model that uses a web module. The maximum number of simultaneous sessions is limited by the ability to handle parallel threads in the Java runtime environment. The maximum number of sessions constrains the number of users in a model and this will again limit the throughput. To reduce this constraint the FTP interface module was developed where the web pages and files are simply downloaded into memory and discarded without further inspection. Hence, the downloading of pages and files will be much less computer demanding because the GenSyn program now only needs to send a request for a page/file.

The throughput has been studied by use of the FTP model as described in Section 2.2.2.1 with

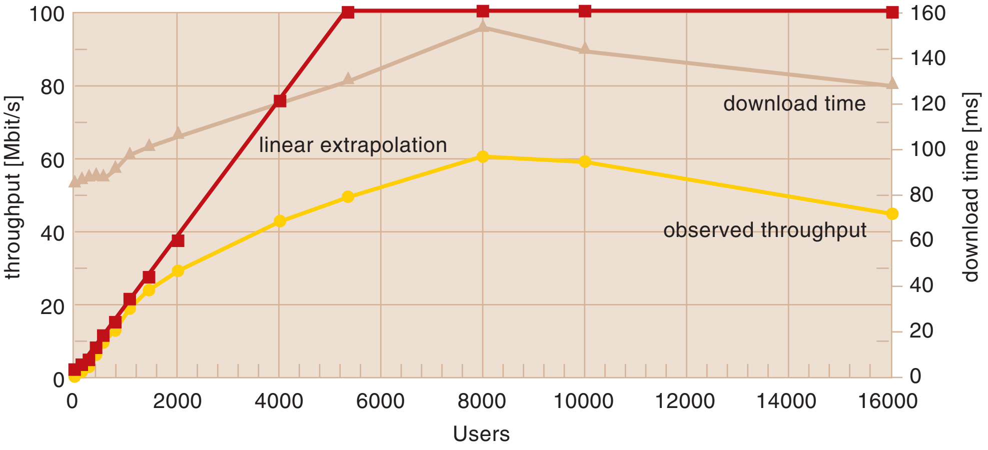


Figure 13 The constraints of a GenSyn process downloading files to a single machine (average file size = 154 kbytes, average download time = 86 ms, interface rate = 100 Mbit/s)

a different number of users. In a lightly loaded scenario, i.e. where the number of sources in the FTP models is small, it is very rare that more than one simultaneous session is observed. In that region, it is expected that the throughput is increasing linearly with the number of users. In Figure 13 the observed throughput is plotted as a function of the number of users and compared with a linear extrapolation of the increase observed in the lightly loaded region. When the load increases several simultaneous downloads will occur and resource conflict, packet loss and reduced throughput are observed. With simultaneous sessions the average download time per page/file will increase. In Figure 13 it is observed that when the download time increases, due to simultaneous sessions or network congestion, the throughput increase is less than linear. From the linear extrapolation it is observed that at approximately 5320 users the average offered traffic is > 1. The TCP window mechanism is activated and starts to regulate the rate for a much lower load level and number of users than this. To identify the exact point where the TCP window mechanism is activated, it is necessary to study e.g. the packet loss ratio.

The plot in Figure 13 shows that the throughput decreases when the number of users is greater than 8000. This is an indication that too many threads are generated by the GenSyn process and the processor capacity and memory size constraints will prevent the creation of new threads with fewer downloads as the result. This is confirmed by the average download time that is reduced in the same region which indicates that there is less network congestion, and by looking at the number of downloaded files during the experiment period – it is increasing up to 8000 and then decreasing.

Finally, observe that a single instance of GenSyn was able to model 8000 FTP clients that generated (received) an average TCP load of 61 Mbit/s through a 100 Mbit/s interface.

The number of users modelled by one GenSyn process running the FTP model is constrained by:

- *Size of memory* for temporal storage of downloaded files, and handling simultaneous threads;
- *Interface card* and network equipment (hub/switch/router) that will cause traffic congestion and increase download times and decrease throughput;
- *Processing time* for managing the dynamics of the finite state machine;
- *Transmission capacity* between FTP client and server;
- *Server performance* where the download files are located.

On a moderate PC configuration (600 MHz Pentium III, 512 Mbytes RAM, Fast Ethernet interface) the TCP throughput is constrained by the network or interface card. Hence, the TCP window mechanism reduces the throughput due to congestion (the offered load is greater than the capacity) before the handling of multi-threads becomes a problem.

2.3.1.2 UDP Traffic – the Packet Generation Constraints

The processing involved with downloading the web pages and files are moderate compared with transmitting UDP packets as required by the CBR, MPEG, and VoIP interface modules. Several experiments have been conducted to get more insight into how the memory size and processing capacity constrain the performance of GenSyn. The network capacity is not an issue for UDP models because GenSyn transmits packets to a specific IP address without considering the congestion situation inside the network, or in fact it does not even require a route to the host[2].

The results from a few experiments conducted by a student [And01] are given in Figure 14[3]. The number of packets sent per second (pps) is plotted for a different number of users in a GenSyn model running on a single machine. For all three model examples it is observed that the packet rate grows linearly as the number of users increases up to a certain point where the handling of threads in the Java runtime environment becomes too heavy for the processor (Pentium III 600 MHz), see Figure 14 for details. The size of the memory (512 Mbytes) is sufficient and will not limit the number of parallel threads. The unconstrained curves are determined by finding the expected number of users in the *Connect* state, and calculating the packet stream rate generated by each interface modules.

[2] *Some problems have been observed with ICMP messages from the destination machine stating "port unreachable". These messages were ignored in Windows, but had to be filtered out in Linux.*

[3] *Each point in the plots is from a single experiment. However, replicated experiments showed that the variance was very small.*

It is important to emphasize that all observations indicate that it is the processor capacity that limits the packet rate (pps) generated from a single instance of the GenSyn process. Increasing the packet size, constrained by the socket buffer size, or the interface card, can increase the bit rate generated. In the case of VoIP in (c), the pps is plotted for different numbers of users for two models of an 8 kbit/s voice channel. In the first case, 10 byte packets sent every 10 ms, it is observed that the GenSyn process reaches its limit at approx. 8500 pps (2.5 Mbit/s incl. 28 bytes overhead) with 3000 users on one machine. The similar observations for the second case where 100 byte packets are sent 100 ms apart, the limit is 2400 pps (2.4 Mbit/s incl. 28 bytes overhead) with 8000 users. The processor capacity is the critical constraint. The exact number of users specifying the knee point is dependent on the hardware the GenSyn process is running on.

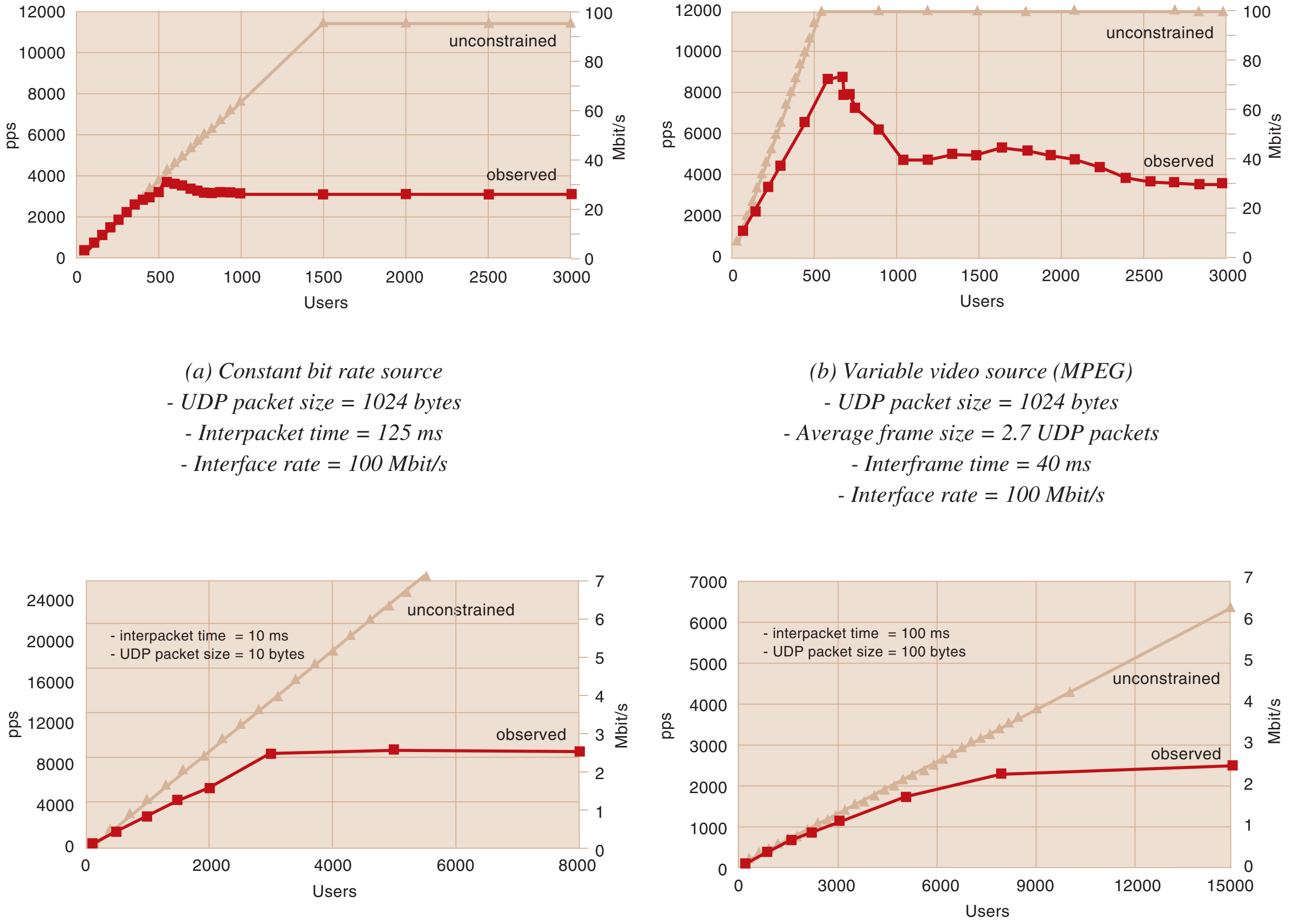
It is observed that the activity rate in the finite state machine (the number of state transitions per time) influences the packet generation efficiency. For example by decreasing the activity rate in a two state CBR model by a factor of ten, the packet rate constraint for a large number of users was increased from 3000 to 4500 packets per second.

Summing up, the major constraints while running a UDP model are

- The handling of simultaneous threads;
- Processing time for managing the dynamics of the finite state machine.

3 A Distributed Test Platform

GenSyn is currently being used for QoS performance testing in an experimental, IP based communication platform that will provide differentiated services. To test its QoS mechanisms, a controllable, and reproducible, mixture of traffic streams with different characteristics is essential. This traffic mixture is generated by GenSyn using the source models currently available and



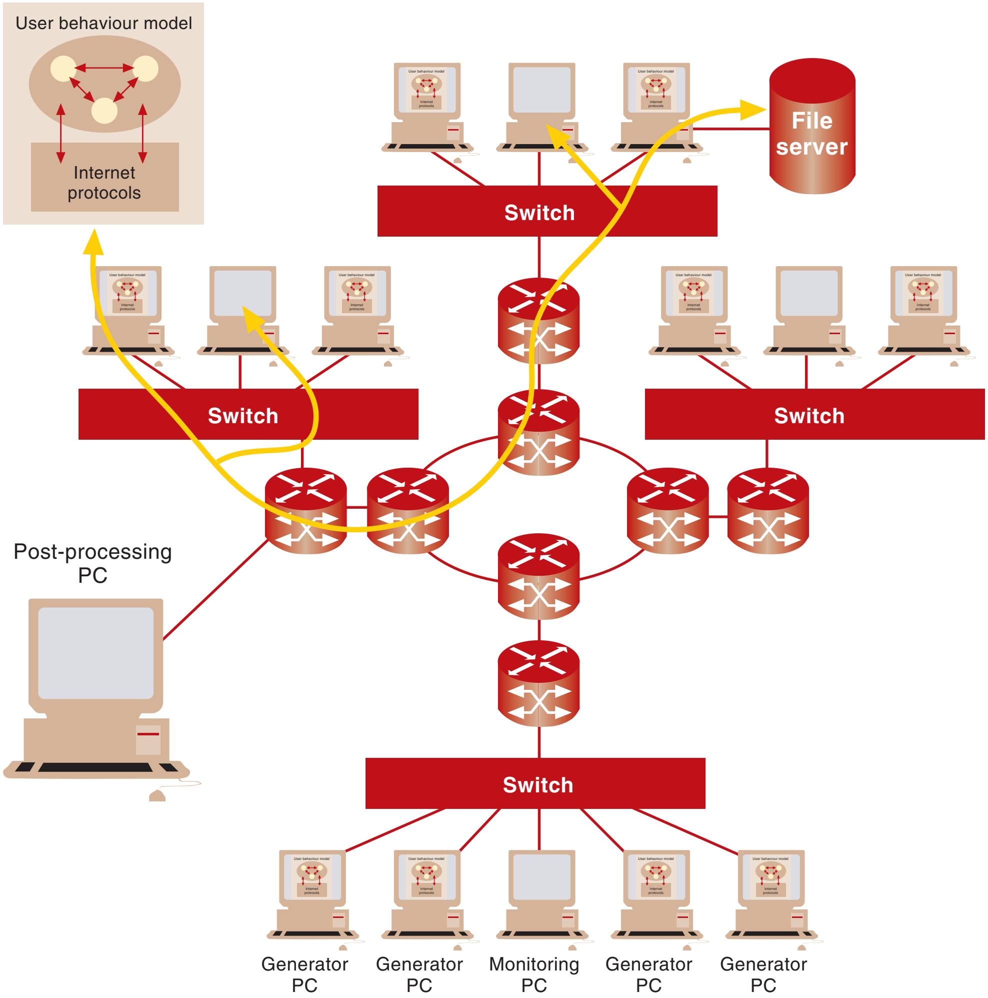
(a) Constant bit rate source
- UDP packet size = 1024 bytes
- Interpacket time = 125 ms
- Interface rate = 100 Mbit/s

(b) Variable video source (MPEG)
- UDP packet size = 1024 bytes
- Average frame size = 2.7 UDP packets
- Interframe time = 40 ms
- Interface rate = 100 Mbit/s

(c) Voice over IP source
- Interface rate = 100 Mbit/s

Figure 14 The constraints of a GenSyn process generating UDP packets from a single machine

Figure 15 GenSyn experiment platform for IP networks



described by the framework of GenSyn. This includes models of web and FTP clients that generate TCP traffic by downloading pages and files from actual web servers, and models that generate UDP traffic from a video server (using MPEG), from voice over IP (VoIP), and in a Constant Bit Rate (CBR) stream.

Besides generating traffic, there is a need for accurate traffic measurements in order to derive performance measures like unidirectional end-to-end packet loss and delay. This has been achieved by deploying PCs dedicated to traffic monitoring using specialized hardware, so-called DAG PCI cards. Section 3.1 describes the GenSyn measurement platform developed for QoS testing of IP networks.

A test scenario defines the application and protocol mixture, load level, traffic matrix and network configuration (DiffServ, Best Effort, etc.). Several test scenarios have been defined for testing of the QoS mechanisms in an IP network. The design of GenSyn test scenarios is presented in Section 3.3 and examples of results that can be obtained from such an experiment are found in Section 3.4.

3.1 An Overview of the Measurement Platform

The GenSyn traffic generator is used in distributed experiments with traffic generators running on several dedicated machines in the network being tested. Because GenSyn is implemented in Java, very little measurement functionality is included on a per IP packet basis. Instead, the measurement platform is based on dedicated monitors, counters at the interface cards (to monitor the routing and load balance), and specialized equipment for generation and analyzation like Smartbits.

Figure 15 shows the GenSyn test topology used for QoS testing of an experimental IP network. The figure illustrates all major components including both GenSyn machines and dedicated monitoring machines. The GenSyn PCs execute

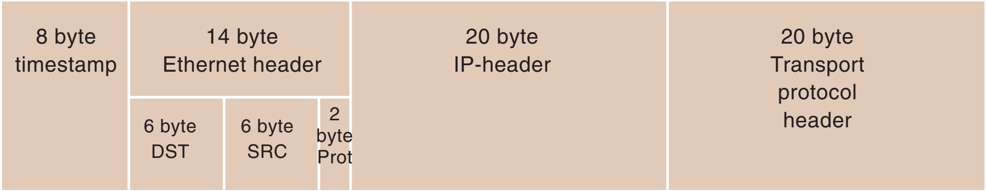


Figure 16 DAG data format over 10/100 Mb/s Ethernet

the GenSyn traffic generator processes while DAG monitors (probes) collect the IP packet traces.

3.2 Using Passive Measurements and DAG Cards

Passive measurement data is collected by observing real packets at selected measurement points. The method captures information contained in the various fields of the packet header. As opposed to active measurements, this approach is non-intrusive and ideally the measurement process does not disturb the operation of the network. Unlike active measurements, passive measurements can gather detailed information about every packet by tracing (taking a copy of) every packet sent and received. The obvious drawback of this method is that the collection of raw packet traces from high capacity networks creates huge data volumes.

The measurement instrumentation for QoS testing of IP networks deploys dedicated PCs placed at strategic locations that passively collect measurement data. These PCs are synchronized by GPS. The motivations for using dedicated monitors with specialized interface boards to passively capture synchronized packet traces were as follows:

- The measurements should not interfere with the traffic generation or the operation of the network being tested.
- Highly accurate timestamps are needed to measure unidirectional delay precisely.
- The monitor must be able to capture packet traces without losing any information even at a high load.

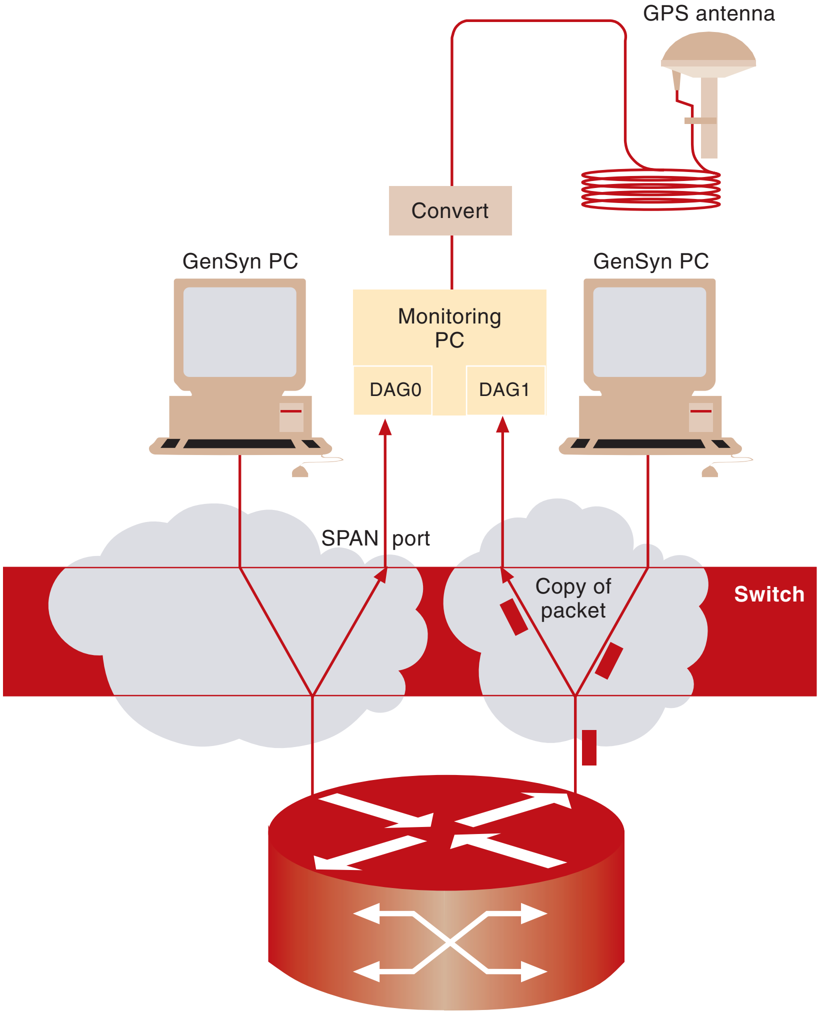
Optionally, each traffic generator could have run software like *tcpdump* to capture packet traces. However, this approach has several limitations including:

- The generation of timestamps in software is inaccurate;
- The measurements would impact the traffic generation;
- The packet capturing software (tcpdump) has been observed to lose information when the load is high.

Each monitor PC has two DAG3.2E Fast Ethernet interface boards [Dag] specialized for capturing packet traces. The DAG cards generate a 64 byte record for each packet received. The record contains Ethernet, IP and transport layer header information together with a timestamp as illustrated in Figure 16.

Packets are tapped from a Fast Ethernet interfaces to a DAG interface board using an Ethernet switch, as shown in Figure 17. The clocks of the DAG interface boards have a very high clock precision and are synchronized by GPS receivers. Hence, synchronisation of the timestamp clocks in the microsecond range is achieved.

Figure 17 Configuration for attaching traffic generating and monitoring PCs



The DAG monitors are located such that every packet sent and received by the GenSyn traffic generators are captured. The location of monitors is shown in Figure 15. This instrumentation enables accurate measurements to be taken of necessary performance metrics like throughput, unidirectional delay and loss.

3.3 Test Scenario Example

In order to describe the main deployment of GenSyn, this section will describe a complete scenario where the objective is to test the QoS of various service classes under different QoS support strategies. The focus is on the end-to-end performance, particularly for the real-time classes. Several traffic scenarios should be defined for testing of the QoS mechanisms in an IP network. In this section the different components of the scenarios are described:

- Type of application (Web, FTP, VoIP, MPEG, CBR, etc.);

- Network configuration (best effort, differentiation);

- Protocol mixture (UDP/TCP ratio);

- Load level;

- Routing (balanced or single bottleneck).

3.3.1 The GenSyn Application Models

The following models are currently available in the GenSyn framework.

TCP traffic – adjusts to the network performance (slow-start window mechanism)

- *Web* – a model of users (clients) that download web-pages with all their content (inclusive applets and images) from real web servers all over the world. The url addresses are found in a parameter list of predefined addresses that may dynamically be updated as the experiment evolves.
- *FTP* – a model of users (clients) that download real files from a server. The files are specified in a parameter list of files.

UDP traffic – no network performance adaptation at transport level

- *VoIP* – a model of the information/media stream from VoIP users. It sends a deterministic stream of packets (fixed size and inter packet arrival time) from each of the active users. The model does not include the call set-up and disconnection phases.

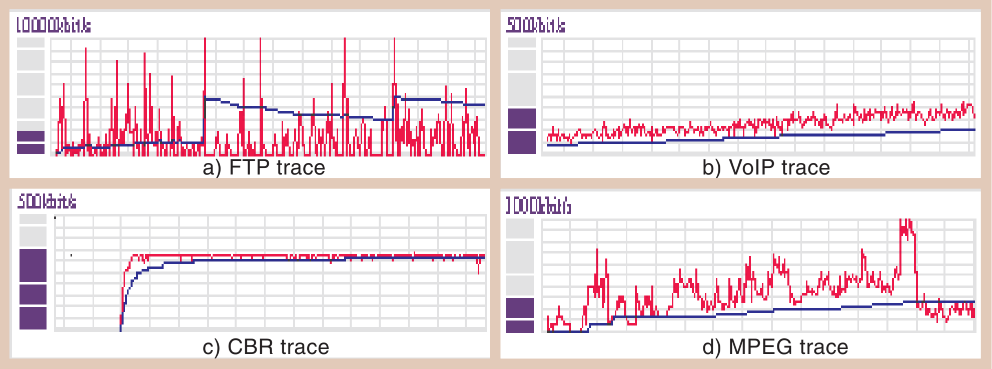
- *MPEG* – a model of a video server that is sending MPEG-1 coded video sequences [Rose95]. All video frames are converted to a number of fixed sized IP packets sent back to back. The interframe distance is a parameter with a default value as recommended by the MPEG-1 codex standard. The clients are implicitly modelled only as incoming requests.

- *CBR* – a model of a multiplex of deterministic streams of packets with phase shifts.

New models can be defined on request, and the current models can easily be changed if this is requested. In Figure 18 snapshots from the GenSyn visualizer are included. The figure illustrates that GenSyn can generate packet flows with very different traffic characteristics.

3.3.2 QoS Mechanisms and Service Differentiation

There is a trend towards building communication platforms for integration of a great variety of applications with different traffic characteristics and users with different Quality of Service requirements. There is a trend in networking towards Full Service Network, i.e. a network for all types of services. The various services and applications need to be treated differently, and hence some means for service differentiation is required with different support for traffic management and control. In an IP based network such differentiation and management techniques are still a research topic. The proposed, and



a) FTP trace

b) VoIP trace

c) CBR trace

d) MPEG trace

Figure 18 Snapshot of 4 traces captured from the visualizer in GenSyn

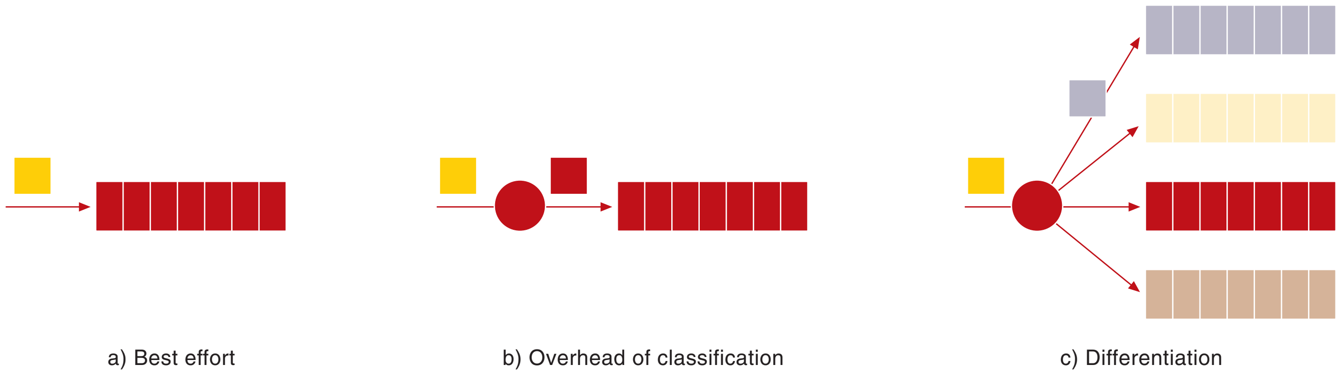


Figure 19 Support for QoS in test

implemented, mechanisms and methods (e.g. DiffServ) need to be studied carefully. Several approaches can be chosen for differentiation of services, e.g. according to

- *Real time requirements* – no (best effort), weak (audio/video streaming), and hard (telephony, interactive video); or

- *Willingness to pay* – economy (free/cheap, only connectivity requirements), business (inexpensive, minimum guaranteed level on QoS requirements) and first class (expensive, hard QoS requirements).

The service differentiation is part of a Service Level Agreement (SLA) that will exist between end-users, network providers, service providers, and content providers. In order to provide (and observe) different QoS performance, some mechanisms and methods are required in the network. To test the implemented mechanisms of commercial routers it has been proposed to carry out the same experiment under different network configurations, see Figure 19.

- *Best effort* – to establish a reference system;

- *Classification* without differentiation (best effort) – to study overhead of the mechanism;

- *Classification with differentiation* in service classes – to study effect of differentiation.

In all 3 cases the QoS performance is studied separately for all service classes although in case 1 and 2 they all receive the same treatment.

For DiffServ, the service differentiation is in accordance with the IP precedence bit setting. The IP precedence bits are the 3 least significant bits of the Type of Service (ToS) field in the IP header. The GenSyn traffic generator cannot change the IP header because GenSyn operates only on the TCP and UDP protocol layers. Hence, in order to get traffic streams in different service classes while using GenSyn it is necessary to include entries in the access lists in the edge routers. These entries define a mapping from the IP addresses of the GenSyn machines and port numbers that can be manipulated by the GenSyn. The static access list entries are only valid to UDP traffic. The TCP traffic sources use the HTTP that has default port number 80. To enable TCP traffic in other than the best effort class (IP precedence = 0), it is necessary to change the mapping of port number 80 and IP precedence for a specific GenSyn IP address for a given experiment. However, in most of the experiments the TCP traffic will be best effort traffic.

3.3.3 Load Level and Application Mixture

Before the GenSyn processes can be distributed on the PCs and started, it is necessary to decide where to run the various processes. Furthermore, it has to be specified what number of users to run for each type of model to produce the requested traffic mixture and load.

3.3.3.1 The Load Level

All traffic scenarios define their load level as the total offered load from all GenSyn processes relative to the network capacity. As a start, the following load levels can be defined: e = 0.2, 0.4, 0.6, and 0.8. As an option, other load levels can be defined.

3.3.3.2 The TCP – UDP Mixture

Each load level must define different mixtures of TCP and UDP. Different TCP/UDP mixtures are constructed by the use of the application models that are available in GenSyn.

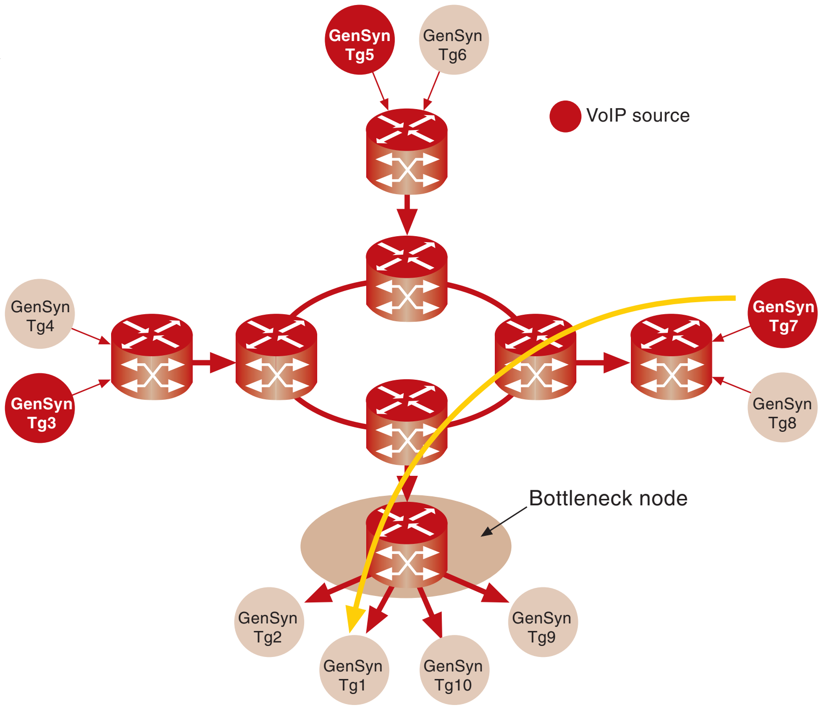
Examples of TCP/UDP mixtures that can be included in a test scenario are:

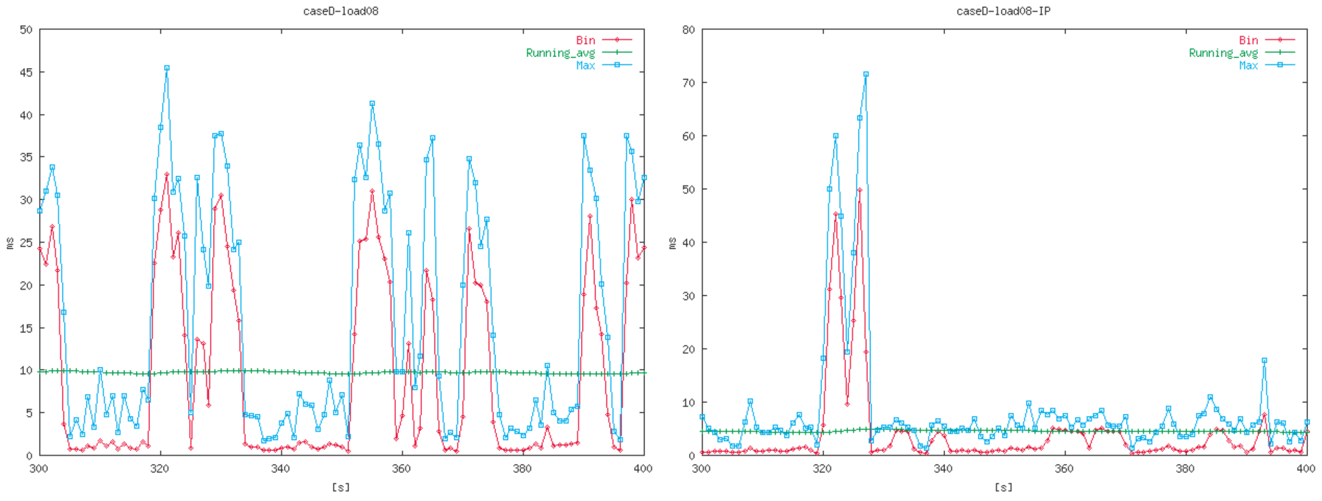
- *Today* – in bytes, 85 % TCP and 15 % UDP traffic;

- *Tomorrow* – in bytes, 40 % TCP and 60 % UDP traffic;

- *Near future* – in bytes, 10 % TCP and 90 % UDP traffic, under the assumption that appli-

Figure 20 Bottleneck scenario – GenSyn machines communicate to create congestion on an edge router

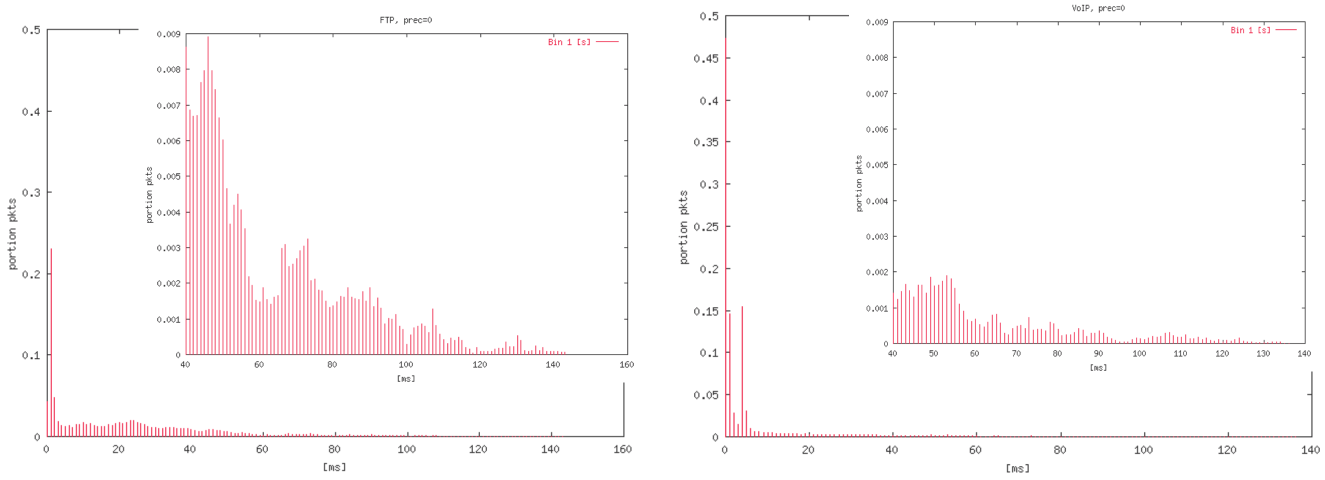




a) Voice as best effort

b) Voice with priority

Figure 21 Delay over time for voice traffic sent as (a) best effort, (b) with priority



a) Voice as best effort

b) Voice with priority

Figure 22 Delay distribution for voice traffic sent as (a) best effort, (b) with priority

cations like video on demand or interactive video become popular.

The latter is expected to be a worse case with respect to TCP performance. The reason is that UDP has no feedback mechanism that adjusts the bitrate according to the congestion in the network similar to what TCP does. This implies that under heavy load, TCP will reduce its sending window to a minimum, while UDP traffic sources continue to send at the same rate. For some applications that are running on top of UDP a rate adaptation or control are implemented, but this is not considered in this example.

3.3.4 Traffic Matrix

Obviously, the routing of packets in an IP network is beyond the control of the GenSyn traffic generators. However, the set-up of an experiment determines which end systems that communicate with one another. Given information about the network topology and routing, it is possible to create e.g. balanced or bottleneck test scenarios. *Balanced* scenarios aim to create an almost equal load on the routers and links in the network, while a *bottleneck* scenario seeks to create a bottleneck in an edge router as shown in Figure 20.

In the bottleneck scenarios, the TCP traffic is generated by the FTP application model running on the GenSyn machines connected to the bottleneck router. Thus, files are downloaded by FTP clients running on these machines from FTP servers running on GenSyn machines connected to the other three edge routers.

Scenarios are tested for various application mixtures, load levels and network configurations. The relative load on various service classes is equivalent for both scenarios.

3.4 Examples of Results

To demonstrate the type of results that can be generated by DataReporter the delay over time (Figure 21) and delay distribution (Figure 22) are plotted for voice in the two cases with and without differentiations.

4 Experiment Support

In an experiment, the configuration, implementation and evaluation of results involve a lot of (manual) work even when the measurement platform is well established and configured. To help the analyst some automated support is under development for setting up GenSyn experiments and post-processing of results.

This section includes a brief description of

- *GenSyn Designer* – setting up a distributed traffic generation and measurement experiment;
- *GenSyn DataReporter* – post-processing of packet traces.

4.1 GenSyn Designer – Setting up an Experiment

The experiences from using GenSyn for QoS performance testing in an experimental, IP based communication platform revealed a need for a support system assisting the analysts in setting up the experiments, collecting data, and post processing it. Furthermore, it is essential to get good support in handling all input and output files that are created and used during an experiment.

For this purpose GenSyn Designer was specified. The designer should provide support for

1 Starting GenSyn processes – create, specify model, set parameters;

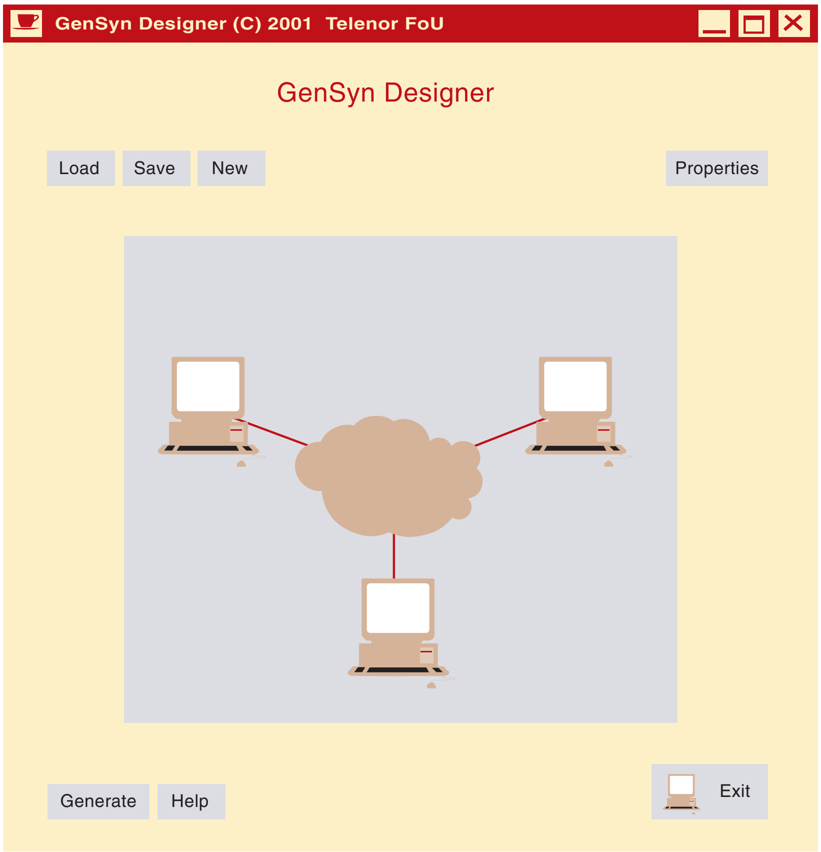
2 Starting measurement probes – define monitors, set filter;

3 Management of experiment – create and maintain a directory of input and output files;

4 Post-processing trace files – apply filters to raw trace data, summarize and plot trace data, see Chapter 4.2.

It is important to emphasize that GenSyn Designer is a scenario editor that allows the ana-

Figure 23 GenSyn Designer GUI: The main window shows the machines in the network



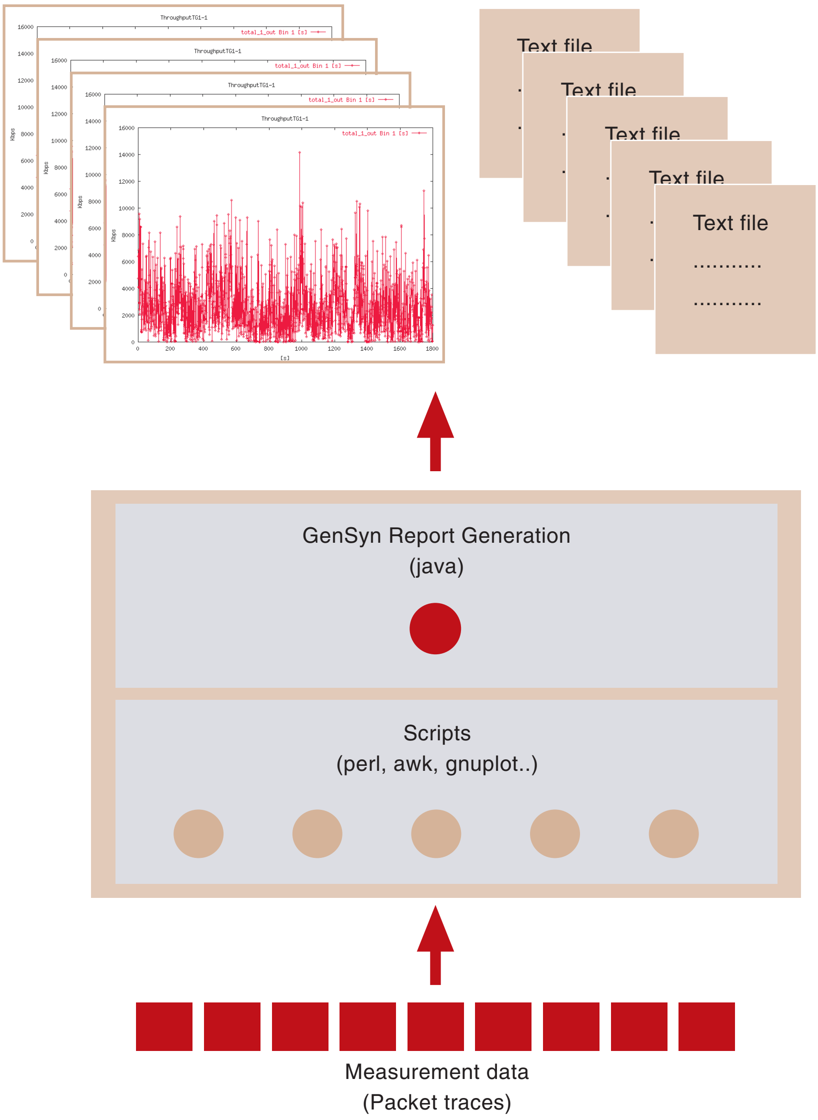


Figure 24 Block diagram of post-processing of measurement data

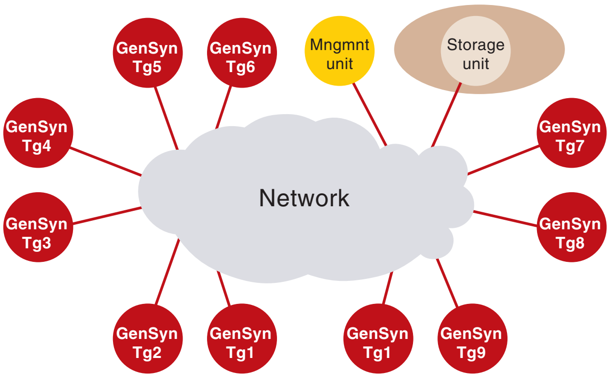


Figure 25 GenSyn experiment

lyst to create and maintain scripts for starting GenSyn, measurement processes and filtering and post-processing of trace data. The Designer is *not* a runtime system in the sense that processes cannot be created, paused/delayed, or monitored. The only support that will be provided is a button that enables the uploading scripts (copying the scripts to the correct machine), and starting all the scripts. As an option, there will be a button for deleting an ongoing experiment, i.e. stop all running processes and delete the trace files.

The GenSyn Designer should provide a graphical user interface for an overview of the machines that run GenSyn, and give useful support in distribution of the GenSyn processes and the measurement probes.

The GenSyn processes require the specification of a number of parameters, a few of them have default values but all parameters should be inspected before starting an experiment. GenSyn Designer should provide the necessary support for the analyst to specify a model for each GenSyn process, the process deployment, and to go through all parameters related to the specific model and the machine.

In the case where GenSyn is not only used for generating background traffic the GenSyn Designer should provide support for instantiation of measurement probes (e.g. tcpdump or DAG on specified machines). The probes can be configured to run various filters with different packet and flow aggregation or selection, depending on the measurement objectives (observation of performance parameters like throughput, loss, delay, jitter, etc.).

GenSyn Designer has specified a hierarchical directory structure that stores the input files, e.g. model types, parameters used, scripts, and the output files, e.g. trace and end report from GenSyn, raw or aggregated trace data from various measurement traces, results from post processing the data, etc.

Figure 23 shows how the first version of the GUI looks like on the screen. It is possible to add and remove machines from the layout. GenSyn Designer can save and load these layouts. Through color encoding the status of specification is indicated, e.g. a machine that should run GenSyn is created, but no model is specified for that machine, or the model is chosen but start and stop times for the process not set.

[4] *It may be noted that the post-processing of packet traces generally is independent of the GenSyn traffic generator. However, currently the GenSyn DataReporter is tailor-made for this purpose.*

4.2 GenSyn DataReporter – Post Processing of Data

The GenSyn DataReporter is a graphical user interface (GUI) implemented in java for post-processing of packet traces. The motivation was to simplify and automate the post-processing of packet traces collected during network tests using the GenSyn traffic generator[4]. The post-processing includes management of input files, input parameters to scripts and output files generated from scripts. Thus, it was seen that a support system was needed. As shown in Figure 25, a central host post-processes the packet traces captured by the DAG monitors or tcpdump. Data reduction on each of the monitoring machines is being considered [EmVi01]. The post-processing involves huge amounts of measurement data and is therefore very computation intensive. The actual processing of measurement data is implemented by using perl, awk and gnuplot scripts. Figure 24 illustrates the software developed for post-processing of measurement data.

Note that the computation for large trace files is rather time consuming. Therefore, the computation is performed only once for each combination of selected options whenever possible by keeping intermediate files.

4.2.1 Centralized Post-processing

The storage unit is a central location where all measurement data collected for a GenSyn test scenario is stored. The measurement data collected for the test scenario includes packet traces (tcpdump or DAG) containing information about packets sent to and from each of the GenSyn machines in the test network (e.g. TG1 – TG10).

The GenSyn DataReporter requires a directory structure as illustrated in Figure 26. That is, before the DataReporter can be run the following steps must be performed:

- Create the directory structure;
- Move rawdata files (DAG or tcpdump traces) to the correct directory;
- Create the GenSyn_machines file with mapping from logical name (and directory) to IP address.

However, a directory structure with several levels is recommended to keep track of various test scenarios. Ideally, the scripts that collect and move measurement data should create the necessary directories and files automatically.

4.2.2 Performance Parameters Computed by DataReporter

The statistics derived from the collected packet traces are classified as point-to-point and multi-point, respectively. Point-to-point statistics are computed by correlating information in two packet traces. Examples of such statistics are average unidirectional delay, unidirectional packet loss ratio and average throughput computed over a bin of a specified duration. Multi-point statistics are defined from observations at a single measurement point (a single packet trace), e.g. number of bytes sent and received to/from various remote machines. Note that the point-to-point statistics provide detailed information about unidirectional performance, while multi-point statistics are less computation intensive but can provide an overview of a given scenario in less time. It may be noted that adding other statistics and graphs is only a matter of changing the scripts and GUI.

4.2.2.1 Multi-point Performance

The performance as observed at a single measurement point is reported. These statistics are computed:

- Average pps[5] and kbps[6] sent to various destinations from a selected traffic generator;
- Average pps and kbps received from various sources by a selected traffic generator.

These statistics are computed over bins of a given duration as a function of time. In addition, various statistics are computed over the entire measurement period.

Note that the multi-point performance statistics only provide an overview as observed at a single measurement point and are not suitable for evaluate end-to-end performance. However, the

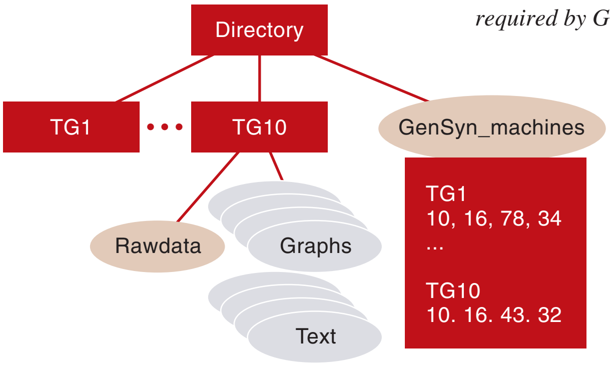


Figure 26 Directories and files required by GenSyn DataReporter

[5] *pps denotes packets per second.*

[6] *bps denotes bits pr second.*

GenSyn availability

A free licence for non-commercial use of GenSyn can be obtained by sending an email to: gensyn@edeber.nta.no. The licence does not include source code.

Please visit http://www.item.ntnu.no/~poulh/GenSyn/gensyn.html for further details.

The GenSyn distribution comes with a set of models and parameter lists with default values. The interface module parameters are described in Table 1.

multi-point performance statistics are less computation intensive and can be useful to check that a certain scenario was run successfully.

4.2.2.2 Point-to-point Performance

The performance from a certain source to a given destination for a specified type of packets is investigated by correlating information from two packet traces. The statistics computed include:

- Average unidirectional packet loss ratio;
- Average unidirectional throughput (pps and bps);
- Unidirectional packet delay
  - Empirical delay distribution in bins of one ms computed over the whole measurement period;

  - Average delay in bins of various sizes as a function of time.

In addition, various statistics are computed for the entire measurement period.

Note that point-to-point performance provides statistics addressing unidirectional performance including delay and packet loss. However, the processing is computation intensive and requires a substantial amount of time for large packet traces.

5 Closing Comments

The measurement platform developed for QoS tests of the IP network consists of two main components; (i) to derive performance measures like end-to-end packet loss and one-way packet delay accurate traffic measurements are conducted by external monitors, i.e. dedicated PCs with a specialized interface card (DAG), and (ii) a Java based traffic generator (GenSyn) that runs on a dedicated PC produces synthetic traffic according to an aggregation of stochastic application models. This test bed configuration is a very flexible platform that opens for doing many exciting and controlled QoS performance evaluation measurements in an IP network.

5.1 Measurement Probes

The actual measurement instrumentation for QoS testing of IP networks deploys dedicated PCs placed at strategic locations that passively collect measurement data. These PCs are synchronized by GPS.

The measurement probes in the test bed are very flexible and enables a capturing level down to a single IP packet if necessary. The monitor set-up can be used with or without traffic generators, the GenSyn is only used when the live traffic in the network is not sufficient or the experiments require controllable and reproducible traffic loads. This means that the concepts developed for this measurement platform can be of interest also in other communication networks, e.g. the service production platform, for testing of stability, scalability, verification of service quality, accounting, billing, and network surveillance systems (OSS).

5.2 Traffic Generators

GenSyn is a Java process that generates IP traffic using a flexible, scalable, stochastic modelling framework for describing the user behaviour of Internet sources. This stochastic behaviour model is linked to the underlying protocol stack and generates real packets into the network. This is a novel modelling approach that chooses the best of two world; flexibility and scalability of composite state models, and accuracy in the protocol behaviour by use of the underlying protocol stack instead of making a model of it.

5.2.1 GenSyn Modelling Framework

The modelling framework itself is flexible and prepared for modelling of many different Internet applications. Several model examples are described in the GenSyn framework, including web, FTP, VoIP, MPEG video, and constant packet rate. The models developed are to some extent parameter controlled and can therefore easily be changed with respect to the number of users, state sojourn times, size of packets, time between packets, IP destination and source addresses, file and web page locations.

New user behaviour state models using existing interface modules can be developed without any changes in the GenSyn implementation.

5.2.2 GenSyn Constraints

The scalability of the generator is limited by the time granularity and time scheduler of the Java Runtime Environment (JRE). The portability of the generator is limited by both the inconsistencies between different versions of Java APIs, and the differences in the time granularity and run-time schedulers of JRE running on different platforms and OS (NT, Linux, Unix). Java is not for real time applications.

Studies of the constraints of a single GenSyn process on a single machine show that

- The number of transmitted UDP packets per second is limited by the processor capacity and the memory size;

- The TCP throughput is constrained by the interface card in the sense that the TCP window mechanism reduces the throughput due to congestion (the offered load is greater than the capacity) before the handling of multi-threads becomes a problem.

5.2.3 GenSyn Measurements

In the current implementation some measurement functionality exists. This includes

- Source and destination ports are added to the UDP packets generated to enable filtering packets by tcpdump;

- Trace file with records of the size of a web page or FTP file, or the length of a phone connection or a video stream;

- Summary report on the total amount of submitted and received data (in bytes and packets), and the number of unsuccessful attempts.

The results in Section 2.3.1 are based on the latter summary report from GenSyn.

In the current version of GenSyn, no end-to-end measurements of real-time performance like delay, jitter (delay variation) and loss are done by GenSyn. These measurements are carried out by the use of trace software (e.g. tcpdump) on dedicated or separate machines. Extension of the built-in measurement functionality in GenSyn is not realistic because of the time granularity problems of Java. If GenSyn needs to process each packet, e.g. add time stamp, sequence number, change TOS bit, this will significantly reduce the performance of the traffic generator. The solution is to make a platform dependent function (in hardware or at least OS dependent) that processes each packet. However, this is in conflict with the philosophy of GenSyn that has portability as a major requirement.

5.2.4 GenSyn Deployment

The GenSyn has been applied for the testing of stability under establishment of an IP network. GenSyn is also an important component in a large-scale testbed consisting of traffic generators and external measurement machines on an IP based test network. The testbed is prepared and used for testing of various QoS design and end-to-end performance of real-time applications like voice over IP and video and TV distribution.

GenSyn has also been used in combination with embedded load generation and measurement equipment like SmartBits [SBit] where GenSyn provides the background load of controllable and realistic elastic load (TCP connections).

5.2.5 Ongoing and Planned Work

Currently, a lot of work is being done on GenSyn and more is planned in the near future. The key issues are:

Extend the model template library – In the current version of the generator there are interface modules to support the download of web pages and files through http, video streaming, and VoIP and constant packet rate. This library will constantly be extended as new requirements appear. In the licence agreement that accompanies the GenSyn distribution, the licensee is invited to return to the distributor all models that are developed using the GenSyn framework.

Validation and verification – The correctness of GenSyn models relative to the models defined is verified, see [HeLu99]. The traffic stream from the models defined in the GenSyn framework is studied and one example was given in this paper. More work on checking the validity of the models and develop new models needs to be done.

Network measurements – GenSyn is now being deployed in a fully equipped IP platform with DiffServ and MPLS functionality and several different applications. The measurements from these experiments will demonstrate the applicability with respect to generating realistic traffic, and will serve as a verification of the GenSyn process.

The GenSyn is a Java process that generates IP traffic using a flexible, scalable, stochastic modelling framework for describing user behaviour of sources. This stochastic behaviour model is linked to the underlying protocol stack and generates real packets into the network. This is a novel modelling approach that chooses the better of two world, flexibility and scalability of composite state models, and accuracy in the protocol behaviour by use of the underlying protocol stack instead of making a model of it.

References

[And01] Andreassen, T R. 2001. *Controlled generation of Internet Traffic.* Trondheim, Norwegian University of Science and Technology (NTNU), Dept. of Telematics.

[Ake99] Arvidsson, Å. 1999. On traffic models for TCP/IP. In: *Proceedings of the 16th International Teletraffic congress, ITC'16,* Edinburgh, UK.

[BaCr98] Badford, P, Crovella, M. 1998. Generating representative Web workloads for network and server performance valuation. In: *Proceedings of ACM SIGMETRICS'98,* Madison, WI, USA, 151–160.

[Brad69] Brady, P T. 1969. A model for generating on-off speech patterns in a two-way conversion. *Bell system technical journal,* 48, 2445–2472.

[ChLi99] Choi, H-K, Limb, J O. 1999. A Behavioural Model of Web Traffic. In: *Proceedings of the 7th International Conference on Network Protocols (ICNP'99),* Toronto, Canada, 327–334. (Extended version: http://users.ece.gatech.edu/~hkchoi/paper/model.pdf)

[Dag] Graham, I D et al. 1998. Nonintrusive and Accurate Measurements of Unidirectional Delay and Delay Variation on the Internet. In: *Proceedings of INET'98.* (http://www.comms.uab.es/inet99/inet98/6g/6g_2.htm) (Dag project homepage http://dag.cs.waikato.ac.nz/)

[Dan92] Danzig, P B et al. 1992. An empirical workload model for driving wide-area TCP/IP network simulations. *Internetworking: research and experience,* 3, 1–26.

[Heeg00] Heegaard, P E. 2000. GenSyn – a Java based generator of synthetic Internet traffic linking user behaviour models to real network protocols. In: *Proceedings from ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management,* Monterey, CA, USA.

[Heeg01] Heegaard, P E. 2001. GenSyn – a Java based generator of synthetic Internet. Submitted to *Advances in Performance Analysis*.

[HeHo95] Helvik, B E, Hofseth, L. 1995. Self-similar traffic and multilevel source models. In: *The 12th Nordic Teletraffic Seminar (NTS-12).* Norros, I and Virtamo, J (eds). Espoo, Finland, 407–420. VTT Information Technology.

[HeLu99] Heegaard, P E, Lu, M. 1999. *GenSyn – Java based generator of synthetic Internet traffic.* Trondheim, SINTEF. (SINTEF Technical report STF40 A99078.)

[Helv95] Helvik, B E. 1995. Synthetic Load Generation for ATM Traffic Measurements. *Telektronikk*, 91 (2/3), 174–194.

[HMM93] Helvik, B E, Melteig, O, Morland, L. 1993. The synthesized traffic generator; objectives, design and capabilities. In: *Integrated Broadband Communication Networks and Services (IBCN&S).* IFIP, Elsevier, Copenhagen, Denmark. (Also available as SINTEF Technical Report STF40 A93077.)

[LTWW94] Leland, W E et al. 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transaction of Networking,* 1–15.

[MTW99] Miller, G J, Thompson, K, Wilder, R. 1998. Performance Measurements on the vBNS. In: *Proceedings of Interop '98,* Las Vegas, NV.

[Rose95] Rose, O. 1995. *Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems.* University of Wurzburg, Institute of Computer Science. (Technical Report 101.) (The traces obtained from: ftp-info.informatik.uni-wuerzburg.de/pub/MPEG.)

[SBit] *SmartBits info.* (2001, June 6) [online] – URL: http://www.netcomsystems.com/

[Vic98] Vicari, N. 1998. *Models of WWW-Traffic: a Comparison of Pareto and Logarithmic Histogram Models.* University of Wurzburg, Institute of Computer Science. Research Report Series. (Report No. 198.)

[ViEm01] Viken, B Å, Emstad, P J. 2001. Traffic measurements in IP networks. *Telektronikk*, 97 (2), 230–244. (This issue.)

[WTSW97] Willinger, W et al. 1997. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM transactions on networking,* 5 (1), 71–86.

Methods for Monitoring, Controlling and Charging QoS in IP Networks

JORMA JORMAKKA, IRENA GRGIC AND VASILIOS SIRIS

Jorma Jormakka (45) is a Professor in the Networking Laboratory of Helsinki University of Technology (HUT) and in the Technical Institute of the National Defense College of Finland. His current research focuses on protocols, communication software, services, security and QoS issues. He holds a PhD in mathematics from the University of Helsinki and an MSc in Electronics Engineering from HUT. He was the task leader of Task 4 in EURESCOM P906-GI.

jormakka@tct.hut.fi

Irena Grgic (30) received her BScEE and MScEE from the Faculty of Electrical Engineering and Computing at the University of Zagreb, Croatia, in 1996 and 1999, respectively. She is currently working at Telenor R&D as a Research Scientist within the Future com Business programme. Her main areas of interest include Quality of Service issues, Service Level Agreements handling, management, negotiations, as well as pricing and charging for IP-based services provided end-to-end in a multiprovider environment. She has participated in several international projects handling these topics, among them EURESCOM P906-GI.

irena.grgic@telenor.com

1 Introduction

Internet Protocol (IP)-based networks and TCP/IP-applications are steadily getting more popular. Many applications require more than best-effort service, i.e. they need Quality of Service (QoS) guarantees. Studies of customer expectations [Alt00] indicate that users would like service differentiation. Today's networks cannot support different requirements coming from different applications, since the methodology, mechanisms and their implementations are not yet mature. Some mechanisms assuring service differentiation and QoS support in IP-based networks are available, like the Integrated Services (IntServ) and the Differentiated Services (DiffServ) standardised by IETF, but it is not clear how to use these in order to deliver the service end-to-end with the quality agreed with a user. Nor is it clear which parameters to use to express quality at the application service level, i.e. the quality the user can directly perceive. Moreover, mapping these parameters to network performance parameter is not a trivial task. In addition, no services are charged for the quality they are provided with.

The EURESCOM project P906-GI QUASIMODO (Quality of Service Methodologies and solutions within the service framework: Measuring, Managing and Charging QoS) tried to answer some of these issues. The main idea of offering a reasonable set of quality classes to users according to their needs and possibilities (e.g. to pay) was investigated by developing and implementing the QUASI-model.

2 The QUASI-model

The QUASI-model [P906-1] was intended as a simple and practical way to offer several classes of service to customers. As a basic assumption, it was decided that guarantees could only be offered within the network under provider's control, i.e. between certain edge routers called Measurement Reference Points (MRPs). In order to provide end-to-end quality to the user, the user's domain (e.g. LAN, CPE, applications) has to be characterised, i.e. their contribution to the overall quality has to be taken into account. Therefore, in the simple scenario depicted in Figure 1, it is assumed that the provider can offer a service directly to the end-user and control the network portion between MRPs (A and B). Moreover, the provider would characterise the user's domains in terms of describing minimum system characteristics and performance requirements (points C – A; B – D). The users are connected to the provider's network through an access network – only the LAN access was investigated in the project, but with modifications the QUASI-model can be applicable to other access networks. The users are assumed not to be very literate in technology and QoS in particular, implying they are not supposed to set any mechanisms themselves. On the contrary, they should get QoS as offered from the operator, who can only guarantee quality between the MRPs.

During the project it was understood that this service cannot be offered directly to a user, since the guarantees were expressed in terms of Network Performance Level (NPL) parameters, i.e. delay, jitter and loss. Therefore, a QUASI-aware business model with a role of a Service Provider (SP) added, was introduced.

A business model, in general, describes different roles involved in service provisioning, and their corresponding relations. By role is assumed a set of activities a business organisation (or an actor) can perform in order to produce/consume a service. Different roles exchange information and have relationships. Some examples of roles are: user, customer, vendor, service provider, net-

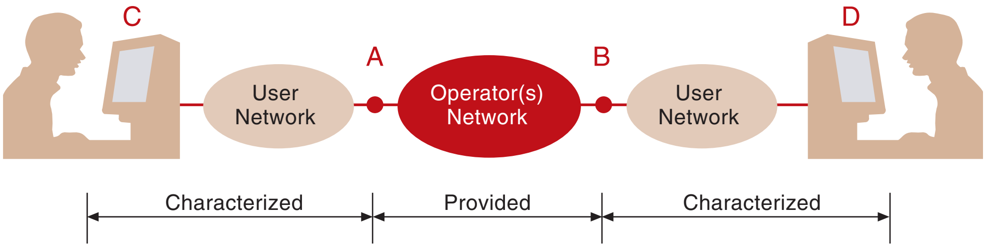


Figure 1 Original QUASI-model physical scenario

Vasilios Siris (34) is a Researcher at the Institute of Computer Science of the Foundation for Research and Technology – Hellas (FORTH), and visiting Professor at the Dept. of Computer Science of the University of Crete, Greece. His research interests include measurement and analysis of network traffic, flexible charging schemes based on resource usage for Service Level Agreements, and service differentiation using weighted end-to-end congestion control mechanisms. Dr. Siris obtained his PhD in Computer Science from the University of Crete, in 1998, his MS degree in Computer Science from Northeastern University, Boston, USA in 1992, and his BS degree in Physics from the University of Athens, Greece in 1990.

vsiris@ics.forth.gr

work operator, content provider, content hosting provider, retailer, application service provider, etc. One role can be played by many actors and it is common for an actor/organisation to play several roles, e.g. the roles of a network provider and a service provider, the roles of a customer and user, and so on. An actor should be considered as a company playing a set of roles on the market.

The business model considered in the project is depicted in Figure 2. The model illustrates the relations between various roles, i.e. SP, Network Operator (NO) and user. Each business relation may involve some form of the agreement between the corresponding parties. For simplicity, it is assumed that the user also plays the role of the customer, and the number of roles is limited so it can be focused on the issues relevant for the QUASI-model. Furthermore, in the general case, the SP can have business relations with other SPs, and with more than one NO. The same holds for the NO, which can have business relations with other NOs.

The relationship between a user and a SP is that of a consumer-supplier model. A similar relationship, but on a wholesale basis, exists between the SP and the NP. The two interfaces shown in Figure 2 include the following:

- User-SP: The service provided over this interface is typically an end service, such as Internet access or access to a particular application (e.g. IP telephony), and is provided on a retail basis. The service to be provided and the description of its quality are described in the Service Level Agreement (SLA) made between the user and the SP. Naturally, SLA contains other information, e.g. pricing, legal, etc. Details of the underlying transport service required for delivering the service might or might not be transparent to the user.

- SP-NO: The NO provides, on wholesale basis, network connectivity services to SP. SP needs these services, since they are necessary for delivering the end service to users. The connectivity service enables the transport of bits, and is typically defined in terms of an SLA which, similar as above, specifies the rights and obligations of both the NO and its user, i.e. SP. The NO agrees to provide a certain NPL to the SP, and the SP agrees on the characteristics of the traffic that he is allowed to send. The latter is given by the traffic profile of the SLA. Other issues (e.g. pricing information, legal issues) could be included in the SLA as well.

Note that the lines in Figure 2 represent a direct business relationship, i.e. flow of business information and payments. The actual delivery of the service can follow a different path – for example, the provisioning of IP telephony to the user by the IP Telephony Service Provider (ITSP) that has a business relationship will be agreed in the SLA, while the actual service delivery will be realised via a network operator providing Internet access.

Having both physical reference and logical business models, the QUASI-model principles are as described in the following. The user requires a certain Quality Class (QC) from the SP for the application/service used. Applications are classified according to their requirements on the IP service (e.g. real time, non-real time, etc.) into Application Categories (ACs). The SP maps QC and AC and gets NPLs necessary to support the user's requirements. An NPL includes a set of NP parameters (NPPs) with related level and guarantees. Then SP has to agree with the NO for provision of the network connectivity service with the performance values as specified in the NPLs for the traffic aggregated per QC. The NO maps the NPLs to adequate quality classes on the network level and applies proper control mechanisms in order to provide the requested performance levels. Naturally, the business between SP and NO has to be described in the SLA made between them for the network connectivity with NPL offered to the user. In addition, the user and SP will make the SLA where the mapping between QC and AC will be stated in the language understandable to the user.

3 QUASI-model Mappings

The relationship between different QCs and ACs, as to be mapped by the SP, are given in a matrix form (Table 1).

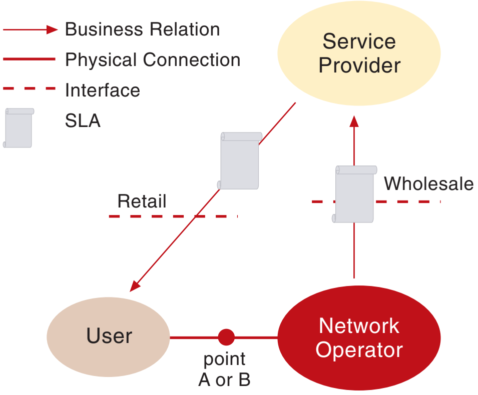


Figure 2 Business model considered for the QUASIMODO project

	AC_1	AC_2	...	AC_n
QC_1	NPL_{11}	NPL_{12}	...	NPL_{1n}
QC_2	NPL_{21}	NPL_{22}	...	NPL_{2n}
...	...	...	...	...
QC_m	NPL_{m1}	NPL_{m2}	...	NPL_{mn}

Table 1 The general QUASI-model

The mappings between QC, AC and NPL definition are described in more detail in the following.

Each pair (QC, AC) in the matrix is called a Quality Class Specification (QCS), and its value corresponds to a *Network Performance Level* (NPL). As briefly mentioned above, an NPL is a (feasible/small) set of NPPs (e.g. delay, jitter and loss), each with an objective, i.e. value[1] and a guarantee per parameter that its value will stay inside the range. By doing so, it is enabled to relate different quality classes with different application categories. In other words, the objectives of an NPL are the target for the network in order to provide the desired QC when an application (belonging to an AC) is instantiated. Each NPL can be seen as a data 'flow' within the network and must be treated/managed accordingly.

Regarding ACs, three options have been chosen: *interactive real-time, non interactive real-time* and *non real-time*. Regarding NPLs, relevant NPPs chosen are *delay, jitter* and *loss*. Regarding their values, upper bounds on the values of those parameters and related guarantees (e.g. % of time the parameter value is below the threshold) are specified.

As input to the experiments (as described in next chapters), a QUASI-model with two QCs (i.e. *Premium* and *Basic*) and three ACs relative to applications characterised by significantly different performance requirements (i.e. non real-time, non interactive real-time, interactive real-time) was used. Combining two QCs with three ACs results in at most six possible data flows to be treated differently in the network in order to preserve the performance requirements of the relative services/applications (Table 2). The Premium class of each AC would be treated better than the Basic class. All the remaining traffic not belonging to these two subscribed quality classes or exceeding the agreed traffic profile or the total bandwidth available, is treated as Best Effort (no NPL values are given for BE).

		Application Categories		
		AC1	**AC2**	**AC3**
Quality	**Premium**	NPL_{11}	NPL_{12}	NPL_{13}
Classes	**Basic**	NPL_{21}	NPL_{22}	NPL_{23}
	Best Effort	–	–	–

Table 2 The practical QUASI-model

In order to implement this kind of model, there should be methods for the operator to monitor the achieved NPL and control the network so that guaranteed NPL is reached.

One of the first things to be fixed was selection of the NPL parameters. The project started by investigating similar architectures. They mostly use end-to-end delay of IP packets, IP packet loss ratio, jitter – which may mean variation of delay between consecutive IP-packets or something else, and throughput in some sense. There are definitions for similar parameters in the ITU-T Recommendation I.380 and IETF's IPPM (RFC2330). Why the QUASI-model definitions are not adopted from either of these is partially caused by the QUASI-model as the parameters should be between MRPs and partially since these standards also define the way to measure them, indicating test traffic based measurements. There was some concern whether measuring low loss ratios with test traffic would be a good solution and whether test traffic could accurately describe delay and jitter of user traffic. The reason why jitter was defined as standard deviation in a small time interval and not from the difference between two consecutive packets was that, if the measurement is on user traffic, the packets may belong to different users and we did not want to measure per flow. These considerations led to the following definitions for the NPL-parameters:

Delay: Delay is the average delay over the time interval of 16 minutes of one-way transfer

[1] *Note that the value may be expressed as a range.*

delays of IP-packets of a given AC, QC from MRP A to MRP B.

Jitter: Jitter is 2*standard deviation of the delay over the time interval of 10 seconds of one-way transfer delays of IP-packets of a given AC, QC from MRP A to MRP B. Jitter is averaged over several time slots of 10 seconds and reported every 16 minutes.

Loss: Loss is the loss probability over the time interval of 16 minutes of IP-packets of a given AC, QC transmitted from MRP A to MRP B.

The choice of 16 minutes was made because we wanted the measurements in the DiffServ implementation to be over the same interval as in the QUASI-IntServ implementation for easier comparison. In the DiffServ implementation measurements were done using CISCO SAA creating test traffic. The time between test traffic packets could not be easily assigned an arbitrary fractional value. We set 2 * 8 = 16 meaning 8 test bursts and 2 minutes between a burst. Each test burst of SAA consisted of a sequence of packets selected to measure the jitter over 10 seconds.

The time interval of 16 minutes may seem far too long considering burstiness of Internet traffic. However, better QoS flows are often assumed to be less bursty, data produced by measurements cannot be very large for storing and processing reasons, and dynamic management is not expected to follow fast changes in traffic volumes but to adapt to longer time variations. In each 16 minute time slot the RSVP-pipes are dimensioned to carry peak traffic and there is admission control for connections and shaping of packets by RED in the edge routers.

If the QUASI-IntServ were used in practice, the time slot size could be set to 5, 15 or 30 minutes, as customary in PSTN. The time slot size is not intended to be in the range of seconds, though traffic variations are very fast in the Internet. The time slot should be on a time scale suitable for QoS monitoring and charging. Congestion control mechanisms resolve problems on smaller time scales. We should note that the QUASI-IntServ implementation is only a test implementation: several aspects, like security, management and accounting should be added or improved before it can be used.

There were different views concerning the measurement of connection blocking – IP does not have connections, but still there are flows for each user – and of something similar to throughput. Throughput for the QUASI-model is something the user requests, but if the user does not get it, then we assume that we see some deterioration in the NPL-parameters and do not need to measure the achieved throughput. This may or may not be true, and there is an optional NPL parameter resembling achieved throughput called the transfer rate.

The usual procedure to set quality requirements to a network is to select the quality parameters, select some reference connections or scenarios and to set target values to the quality parameters. The project did not follow this procedure, but rather chose to run a number of experiments testing users' acceptability. A sample table of upper bounds resulting from a set of tests was compiled (Table 3).

The table contains figures, which can be understood completely after studying the tests as described in [P906-1]. A brief explanation is

Table 3 NPLs and guarantee levels for the QUASI-model (NP values to be considered as upper bounds)

QC	AC	**AC 1** **Interactive** **Real-Time**	**AC 2** **Non Interactive** **Real-Time**	**AC 3** **Non Real-Time**
Premium	Delay	150 msec	300 msec	100 msec
	Jitter	3 msec	50 msec	best effort
	Loss	2 %	1 %	2.5 %
	Guarantee	99 %	99 %	98 %
Basic	Delay	800 msec	600 msec	300 msec
	Jitter	3 msec	100 msec	best effort
	Loss	4 %	5 %	15 %
	Guarantee	95 %	95 %	92 %

given here – e.g. the very small jitter of 3 ms does not mean that the users can detect jitters so small, as jitter sensitive applications have receiver buffers. It is based on observable distortion of the signal and applications with no receiver buffering.

In general, the values given in Table 3 are not a recommendation, but the examples of the values achieved as a result of some tests.

4 Implementation of the QUASI-model

The next step was to implement the QUASI-model monitoring and control mechanisms. Several techniques can be used to implement the QUASI-model. The project focused on two techniques: IETF's DiffServ and IETF's IntServ. We did not consider MPLS – though it may be very interesting – since it was investigated in other EURESCOM projects. We also discarded some more complicated QoS architectures (XRM, HeiRat) as they were rather far from the basic ideas of the QUASI-model, see [P906-3].

4.1 DiffServ Concerns

In this paper we will not describe the DiffServ based implementation, but it is documented in [P906-4]. It supports quality classes by setting the Type of Service (TOS) field in the packets. A user selects a desired service quality for his application by subscribing to it in a WWW-page. This information is inserted to access lists of all edge routers. The edge router where the user's access network is connected marks the TOS-field in the packets. The edge router gets the mapping information from the access lists and therefore the solution also supports the ability to receive good quality, not only to send. QUASI-model NPL classes are given different quality by setting rate limiters, and setting limits as to how much the rate can be exceeded without losses is one way to obtain visibly different quality for different NPL.

QoS monitoring is made using commercial test traffic generating measurement tools, CISCO SAA and some tools to collect measurement data (FireHunter or NetSys). The accuracy is good for delay and jitter, but very small loss probabilities are difficult to measure accurately. Test traffic and user traffic may not be treated equally, e.g. if the user data flow is routed differently. The traffic management part of the DiffServ implementation, notably updating the access lists and rate limiters and their tolerances, needs further elaboration.

The problem of the DiffServ in the QUASI-model was basically a requirement that a subscriber of better quality class should also receive better quality, for instance for videotelephony. One way this could be solved is to extend the application protocols, like SIP, so that they keep knowledge of current connections and insert into access lists all parties of a connection. Such a solution is outside the QUASI-model as it requires improved application protocols. The way this problem was solved in the DiffServ implementation of the QUASI-model was to have all users of quality classes better than best effort in access lists of all edge routers. This solution is not easily scalable. Another scalability problem is the selected way of making class differentiation by carefully adjusting rate limiters to drop more packets from classes with

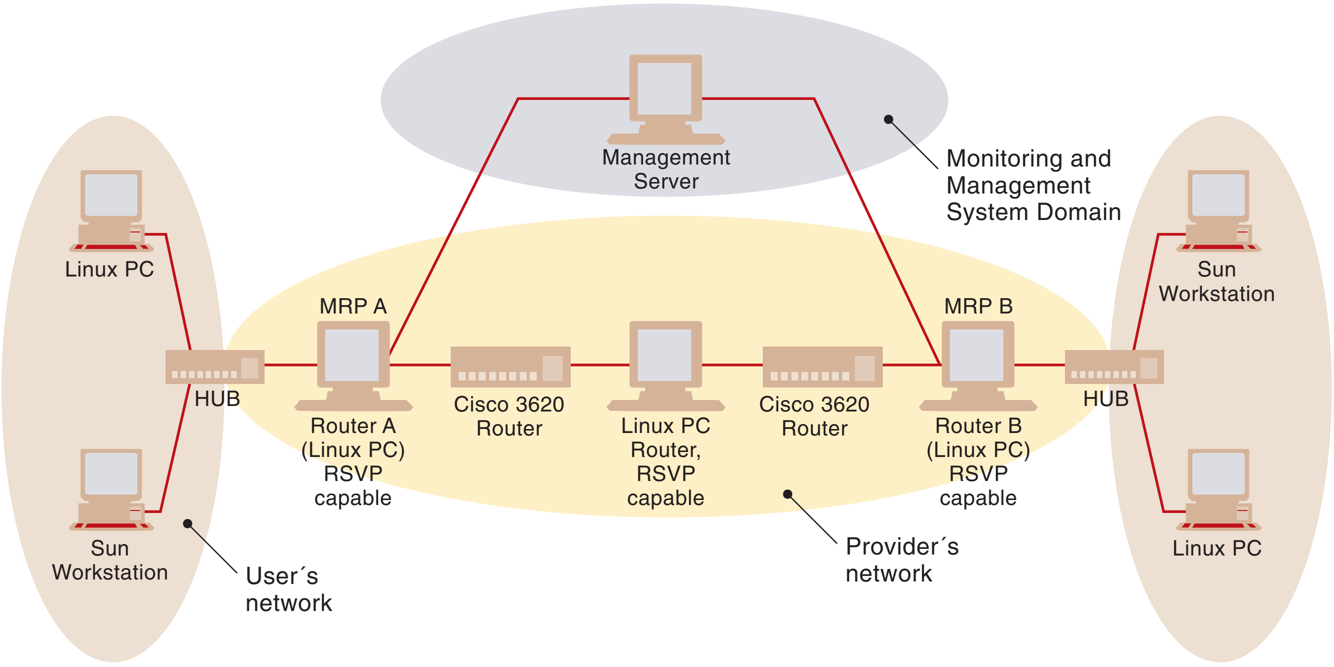


Figure 3 Laboratory test bed

worse quality. This would require an additional protocol so that it could be done dynamically by adjusting the limiters to measured traffic volumes and their burstiness. It is possible to improve the DiffServ implementation in these respects, but these are important scalability problems in DiffServ as a technical solution to the QUASI-model. We should stress that the QUASI-model requirements of class differentiation in some way and ability to receive good quality are very natural from the business point of view for a service of several quality classes. A simple solution to the class differentiation is to offer only two classes and best effort. Then class differentiation should be easy. QUASI-MODO wanted to offer more classes as technically the possible number of different classes is quite large in DiffServ.

4.2 IntServ Concerns

This paper describes the QUASI-IntServ implementation. IntServ is not a natural choice for the QUASI-model: IntServ is end-to-end and the QUASI-model is MRP-to-MRP. IntServ is also considered unscalable. While this opinion may not be permanent, as scalability only needs to be sufficient to the expected size of the network and is affected by performance of RSVP, it is generally thought to be true at least for the near future. Therefore we had to solve the scalability problem and the natural way is to group several RSVP flows together to a reserved pipe. A pipe is similar to a Virtual Path in ATM: VP can contain several Virtual Circuits. Doing this implies that we must add admission control to accept a flow to a pipe. In ATM there is a Source Traffic Descriptor, but we only have a very rough characterisation of sources by the Application Category. It means that the admission control must be rough. How much sense there is in to creating a poor man's ATM in this way can be pondered on, but if an implementation of the QUASI-model is made along IntServ, this is probably the way it would be done.

The laboratory testbed, shown in Figure 3, is quite useful for describing the QUASI-IntServ implementation. There are two MRPs, which in Figure 3 are Linux PCs running RetHat-6.2. The operator's core network is made out of CISCO routers and one Linux PC. It was very good that we included the CISCO routers into the core network, since in this way we had to solve interworking problems of Linux RSVP and CISCO RSVP. Our method of making reservations between MRPs would also have worked better, and we would have overlooked a practical problem, were it not for the CISCO routers. CISCO routers namely decode the RSVP flow from an address in IPv4 and we needed a clumsy address translation, to be explained in Figure 6.

There is also a Linux PC used as a Management Server. It collects NPL measurements from the MRPs and makes the RSVP reservations between MRPs. The user's subnetworks are realised as two Linux PCs and two Sun Workstations running different applications. The network links are made with 10 Mbit/s Ethernet. Implementation of the QUASI-IntServ consists of new software, which was written to the MRPs and to the Management Server. It is only some source files of new software, but inserting the software to the Linux kernel to monitor the NPL parameters required some effort. Figure 4 shows the architecture of this software.

The user agent registers users to the MRP daemon process with a special registration protocol. There is a simple admission control where the MRP daemon estimates from the number of existing connections whether it can accept a new connection to a desired AC/QC. The user can select AC/QC pairs to his application and

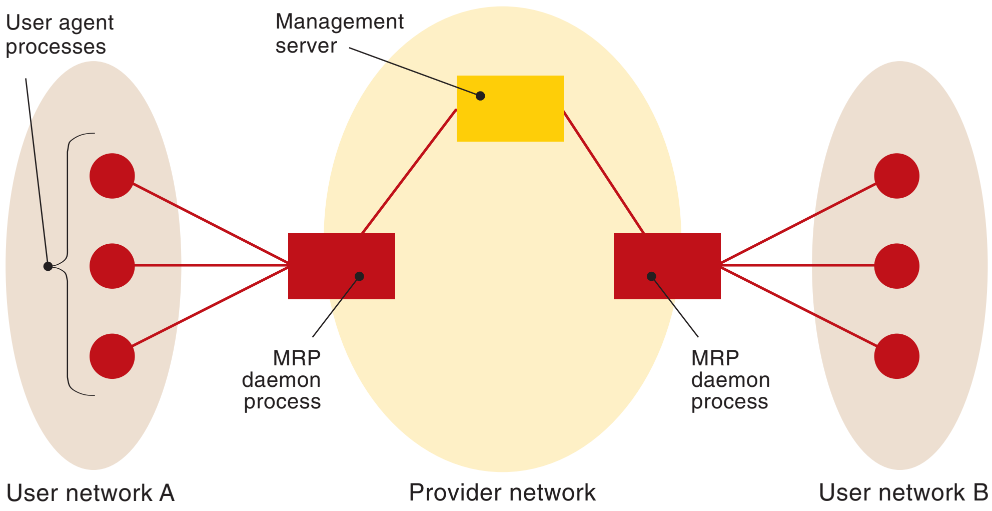


Figure 4 Architecture of QUASI-IntServ software

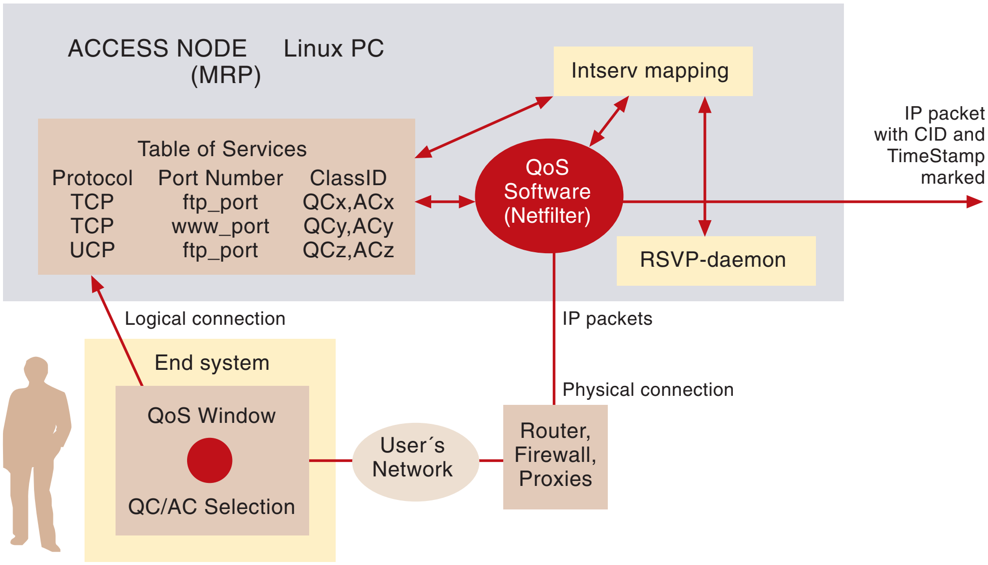


Figure 5 Architecture of the MRP daemon

change it at any time. The MPR daemon process measures the NPL parameters and sends them every 16 minutes to the Management server through a TCP socket interface. The Management server receives the measured values, puts them to a file for the charging application, and calculates with some algorithm new reservations between the MRPs and starts scripts in the MRPs to refresh and to modify the RSVP reservations.

When the network is initialised, RSVP reservations are made between the MRPs. As bandwidth in a practical IP-network is limited and traffic variations could be rather high, we did not set up one-to-one reservations between each MRP for each class but instead configured each MRP to accept up to a given limit traffic coming from all other MRPs. In the laboratory network this difference cannot be seen as there are two MRPs only. After each 16 minute time slot the RSVP reservations are modified by the management server based on the measured NPL values. In the implementation the management server collects the measurement values from each MRP every 16 minutes, but to minimise management traffic we could have sent a trap only if the NPL value measurements show exceptionally bad or good results. There is a small protocol guaranteeing that a user of a better class will also receive good quality, not only be able to send it: the receiving MRP stores the class of each existing connection to a table. Provided that there is a response or reverse flow which sends the correct receiver's class to the sender's MRP, the received quality will also be good.

User traffic must be accepted to a reserved pipe by the MRP, which then must keep tables of users and active connections. There is a simple admission control, but it is currently rough.

Figure 5 shows the structure of the MRP daemon. IP-packets come from a user's application without any quality markings. When these packets enter the MRP, in Figure 5 the Access Node, the MRP daemon looks at a table where for each registered user are kept the assignments of IP packets with given protocol and port number to an AC/QC pair. The AC/QC pair is mapped one-to-one to a ClassID, which is the quality class identifier that the technical solution understands. The packet is marked with the ClassID. The ClassID also gives the MRP number because of the way the ClassIDs are assigned: they are unique to the extent that a receiving MRP can know which MRP sent a packet with a given ClassID. If there is no entry for a given user or (protocol, port)-pair, no marking is done, and the packet will be best effort.

The MRP daemon also sets a time stamp into the IP-packet. This time stamp enables calculation of the delay and jitter from a short term delay variation. We need to time-synchronise the MRPs with some protocol like NTP (Network Time Protocol). As all MRPs are in one operator's network, we can assume that NTP can be used and is sufficiently accurate: 10 ms is a sufficient accuracy to the QUASI-model, and that can be reached.

A packet with a given ClassID is mapped to a correct RSVP reservation selected by the routing mechanism. The reservations are already set in the beginning and modified in a slow pace with updates every 16 minutes, they are not made when a user starts his connection. This is for scalability reasons.

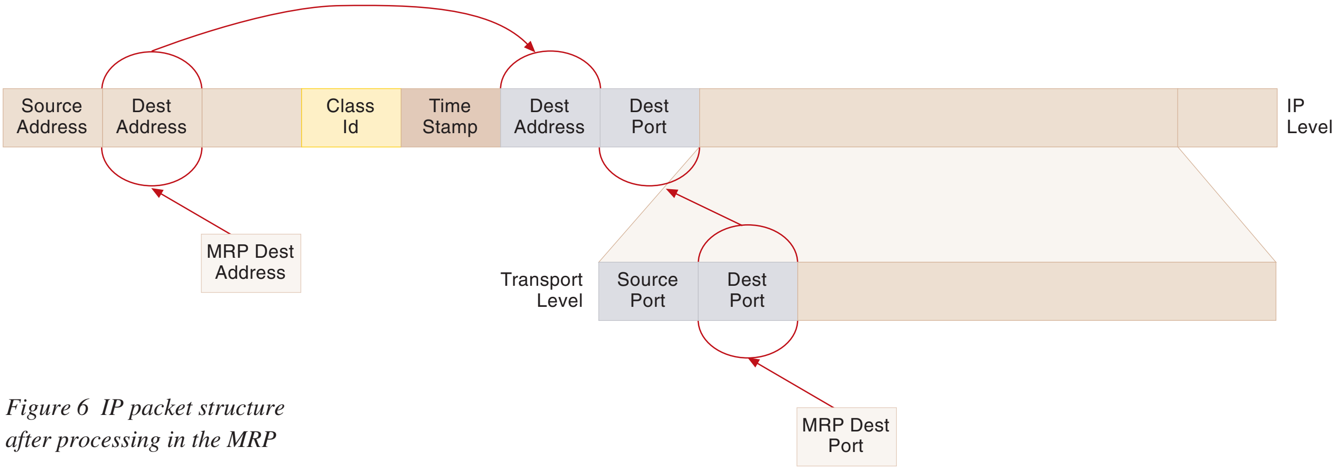


Figure 6 IP packet structure after processing in the MRP

If we were using IPv6, we could set the flowID in the packet and identify the ClassID with the flowID. If we were using only Linux routers, we could decode in all routers the flowID from the ClassID in the IP-packet. As we are using CISCO routers and IP v4, there is a problem: CISCO-routers decode the flow from an address in IP version 4. In QUASI-IntServ RSVP establishes a pipe between two MRPs and the core routers use the address of the user traffic to decide which flow the packet uses. We had to replace the original address by an MRP address and store the original address into the padding field. We use the padding field also for storing the time stamp and the Class ID. In the receiving MRP the original address is restored. This is made with NAT and surprisingly it is rather fast. Figure 6 shows the content of the padding field. There are the necessary time stamp and ClassID, but for IPv4 we have also stored the original addresses there. The receiving MRP restores the original addresses.

The QoS Software converts the QUASI-model Class to network class or ClassID (CID). The IP-packet is marked by setting the CID and the packet is sent forward. If the (protocol, port)-pair is not in the table, no marking is done.

In the implemented version, the user can access the service table and select the AC/QC to his application. A natural way to do this in a real system is to use a WWW-page for user selection of the AC/QC, as was done in the DiffServ implementation. Mapping applications to AC/QC is not an easy matter. We have mapped applications using the port number and the IP-address of packets coming to the MRP. There are special cases that have not been treated: Mbone is IP-on-IP. It is currently not supported. Separating RTP traffic from other UDP traffic, such as DNS, is not treated. The problem is that RTP does not have a protocol number, nor does it have a well-known port number. It may be best to separate other UDP traffic and treat the remaining part as RTP traffic and give it good quality.

The QUASIMODO project made a number of tests to see if the QUASI-IntServ implementation works. The test network is very simple having only two MRPs, and we cannot say if the software actually works in a larger network. It was shown that in the test network it performed fine, and the dynamic bandwidth allocation made by the Management server could adopt the network to changes in traffic load within the time scale of the bandwidth modifications.

One goal of the QUASI-model was to provide several classes which have clear differences in quality. In the QUASI-IntServ implementation RSVP reservations guarantee the bandwidth allocations for AC/QC. They are similar to DiffServ in providing quality for an aggregate of connections, not for an individual user's flow. They are a bit better than DiffServ because if the core network changes, the RSVP reservations are rerouted. We should not have the problem with congestion in the core network. However, do we see class differentiation in this model?

There is a reserved bandwidth for each RSVP pipe and the routers respect the reservations. If the offered traffic to a pipe does not exceed the reserved bandwidth, we will get very good quality for all classes. Users of flexible sources (TCP) see the effect of the bandwidth as data flows slower if there is less bandwidth. Users of unflexible sources (UDP) would see either good quality or bad quality depending on the admission control mechanism. We could overdimension worse quality pipes by admitting more traffic than could be carried with good quality, or we could intentionally drop packets just to make the service differentiation. For QUASI-IntServ it may be better to use blocking in the following way. Best effort class is worse than NPL-classes since the NPL-parameters are not guaranteed. NPL-classes all have good NPL-parameter values and are not differentiated by them. Connection blocking is lower for Premium than for basic and it makes the class differentiation.

In the DiffServ implementation rate limiters are tuned so that more IP-packets are dropped for worse classes even if the total load is low. This method works, but to some extent it is an artificial way to create service differentiation. One should clarify what is the goal of service differentiation and if it might not be better to do it with connection blocking.

5 Measuring Methods

A main motivation for time stamping user traffic instead of measuring with test traffic was to see if time stamping actually is such a performance bottleneck as it has been called. A starting point was a firewall: A firewall is a component you normally would put in places like a router connecting an operator's network to a user's subnetwork. A packet level firewall adds processing load but is normally not considered impossible for this reason. For time stamping we use a rewritten firewall code contained in the Linux OS since the version 2.3x. There is a Netfilter interface by which one can access IP-packets, get them out of the router's queues, modify them and put them back into the queue. In the protocol stack the Netfilter interface is placed between the datalink and network levels in the TCP/IP protocol stack.

Netfilter is basically a set of hooks to packet filtering in the Linux kernel. For IPv4 there are five hooks in different places of routing the packet. One can register to any of the hooks and wait for packets of a given protocol to come, take it for processing and put back. Normally, Netfilter moves the packets to the user space, but if the software is a kernel module, then the packet processing will take place in the kernel, which gives a huge performance advantage over processing in the user space. As we have tried to run the code in the user space and it limited a router's bandwidth considerably, we would recommend writing this kind of modifications as kernel modules, which in our case removed the performance problems. As for time stamping itself, if must get the time in some way. Usually, calling the system clock is slow but fast if the process is a kernel module. One should also note that as the desired time accuracy is only 10 ms, a sufficiently accurate time for a process working at a much higher speed to be made a variable in memory, updated once in 10 ms by a separate process.

5.1 Delay and Jitter Measurement

Measurement of delay and loss are made by inserting time stamps to the padding field of user data packets at the edge routers. This method does not compromise performance and it does not conflict with the use of IPSec or optimisation with MPLS inside the network. It conflicts with fragmentation inside the operator's network and requires less than maximal size packets. It should be possible to avoid these problems as we are talking about one operator's network. Providing quality will have some cost. We use the padding field because that way we can store a time stamp in 16 bits for a resolution of 1 ms to values of up to 65,536 milliseconds. Had we used the IPv4 options field, the time stamp would take 4 bytes and core routers look at this option field as spending time. One bit of the IP header is used to indicate if the padding field contains our data. Currently, we time stamp all packets but only a sample would suffice if performance becomes critical. The sending MRP sets the time stamp using time from the system clock, and the receiving MRP looks at the time stamp and calculates the delay as a difference of the value and the value given by the system clock. Jitter is calculated as a standard deviation over 10 seconds and delay as a longer time average for each time slot in a jumping window manner. Every 16 minutes the MRP daemon writes the values for each AC/QC to a TCP socket, which is read by the Management server.

Tests were made to evaluate both the accuracy and the performance overhead of the delay and jitter measurements. The performance is quite good provided that the code is made a kernel module. Accuracy is very good, the errors are caused by time synchronisation between the clocks and processing time of the time stamping software. The latter was about 4 ms in the tests. It is a value for load, which is not creating a queue. In general, putting some additional processing to a router will give a processing time dependent on the load because of queuing delay, but here the situation is that the software is sufficiently fast to avoid queues and the additional processing time is constant. Tests of jitter showed one interesting thing: we could measure jitter caused by spacing, but test traffic by CISCO SAA between MRPs in the DiffServ implementation would probably not measure this jitter. One should be careful measuring jitter with test traffic; there may be mechanisms which are not affecting test traffic while they affect user traffic.

5.2 Loss Measurement

In the QUASI-IntServ implementation loss is measured by intentionally discarding packets by RED for each ClassID in the MRPs and counting the discarded packets. This rather strange way of loss measurement protects the core and therefore only best effort packets are lost in the core. All losses for better classes occur in the MRPs and as all discarded packets are counted, loss measurement is accurate. We cannot have losses in other routers than the first MRP for any class better than best effort since there is an RSVP pipe and our RED algorithm does not allow more than reserved bandwidth to the pipe. We also have an admission control before the RED, so that if it works perfectly, we would not discard any packets at all. Unfortunately, the admission control cannot work perfectly as we do not have a source traffic descriptor which the source would respect. We must assume that the admission control fails sometimes, but the RED per class will not fail. By monitoring the loss ratio, we can better adjust the admission control also.

Table 4 Management Test Results

	Premium Class (Pr.C.)		Basic Class (B.C)		Processor load (%)	
	Av. Delay (ms)	Av. Jitter (ms)	Av. Delay (ms)	Av. Jitter (ms)	MRP A	MRP B
1st Slot (16 min) **Pr.C – 6 X 512 kbit/s** **B.C – 4 X 256 kbit/s**	6.45	32.96	9.14	33.94	56.17	57.33
	7.98	33.28	9.79	43.37	54.67	52.50
	7.48	35.98	10.61	35.64	50.50	50.83
	8.14	30.24	11.47	40.04	58.00	57.15
	6.38	33.019	9.14	38.70	48.31	58.67
	7.90	29.16	10.06	43.41	52.65	56.33
	7.67	36.15	10.016	23.25	40.15	59.40
	8.56	34.11	11.19	32.73	46.14	58.33
	*** 7.57**	*** 33.11**	*** 10.18**	*** 36.38**	*** 50.82**	*** 56.33**
2nd Slot (16 min) **Pr.C – 6 X 512 kbit/s** **B.C – 4 X 256 kbit/s** **BE – 5 Mbit/s**	10.09	35.20	11.62	40.33	56.00	70.33
	8.08	33.25	10.69	43.42	60.50	65.33
	9.94	32.30	9.75	41.53	63.17	76.67
	8.74	36.34	11.82	43.63	61.83	73.50
	7.53	35.39	12.90	45.74	66.64	58.33
	9.05	34.45	14.98	47.86	70.33	80.17
	8.24	37.50	12.06	48.89	58.67	76.50
	9.06	39.56	15.15	47.00	65.68	81.67
	*** 8.84**	*** 35.50**	*** 12.38**	*** 44.80**	*** 62.85**	*** 66.06**
3rd Slot (16 min) **Pr.C – 6 X 512 kbit/s** **B.C – 4 X 256 kbit/s**	10.99	41.14	18.12	67.17	71.75	87.85
	7.55	35.18	17.15	57.27	82.17	90.15
	8.04	38.23	14.18	64.31	85.45	99.67
	9.66	39.30	15.41	62.35	80.50	63.17
	10.37	38.38	16.61	62.39	87.50	80.83
	8.13	37.47	16.71	59.44	91.67	87.50
	9.93	37.53	17.64	63.48	92.63	90.33
	10.77	38.64	15.58	66.58	87.73	86.33
	*** 9.43**	*** 38.23**	*** 16.43**	*** 65.74**	*** 87.17**	*** 93.99**

For each ClassID the MRP keeps count of all packets which have arrived for a given class and all packets which are dropped for a given class. The ratio of discarded packets and all packets is passed every 16 minutes to the data collecting system. After the time slot of 16 minutes, the counters are zeroed.

The loss measurement was tested and found working. It is accurate by definition and scales very well to small loss ratios. In test traffic methods of measuring loss ratios, like in CISCO SAA Loss measurement, there is a problem since losses are not well evaluated for small loss ratios as only few samples are used. Using more samples increases the load of the loss measurement. We can take an example; in SAA a test call creates 3700 bytes of traffic. If the loss probability is 1% we should require about 10 packets lost in 1000 seconds to be able to get a sufficient statistics, meaning 10 kbit/s of test traffic. If the number of MRPs is 100, then each MRP must generate and receive 1 Mbit/s of test traffic for this one AC/QC. Measuring small loss probabilities, say 10^{-5}, would not be possible with test traffic. Whether IP will never require better loss ratios than about 3 %, sufficient for IP voice, is an interesting question considering the requirements for loss in ATM. In any case the ability to monitor losses should not be the limiting factor.

We expected to see some more differences in test traffic and live traffic measurements, but it turned out that in the test traffic measurements made with CISCO's SAA tool in the DiffServ implementation were sufficiently precise for delay, jitter and for sufficiently high loss probabilities also for loss. There are theoretical cases when differences in test traffic characteristics and user traffic characteristics lead to differences, also when these traffic types are routed differently, but we did not observe anything alarming in this respect.

5.3 Example Test

Seven tests were performed for the measurement and management methods of the QUASI-model. We only present one test here, the test 7 in [P906-5] made by Denis Karpov and Imad Ossaily. This test would verify whether the dynamic re-allocation of RSVP reservations based on NPL measurements works in the network in Figure 3. The test consisted of 3 time slots, each lasting 16 minutes. The tests were made eight times, so Table 4 contains 8 measurement values and their average in boldface. There are two MRP points in Figure 3, and the table shows the delay and jitter values between the MRPs to one direction for two classes of traffic, the premium class and the basic class. Packet losses do not occur for the premium or basic classes. In the test the RSVP allocations are originally set to 3 Mbit/s for the premium class and 1 Mbit/s for the basic class. Then six 512 kbit/s premium class sources and four 256 kbit/s basic class sources are added during a 16 minute time slot. The network adapts to the changed traffic conditions by changing RSVP allocations if the measured values are higher than 95 % of target values. In the second time slot of 16 minutes a best effort source of 5 Mbit/s is added and in the third time slot the best effort source is raised to 7 Mbit/s. The target values for delay and jitter for the premium class are 10 ms and 40 ms, for the basic class the target values are 17 ms and 65 ms. In the test the target values are always achieved. Some service differentiation is seen between the classes seen on high traffic load.

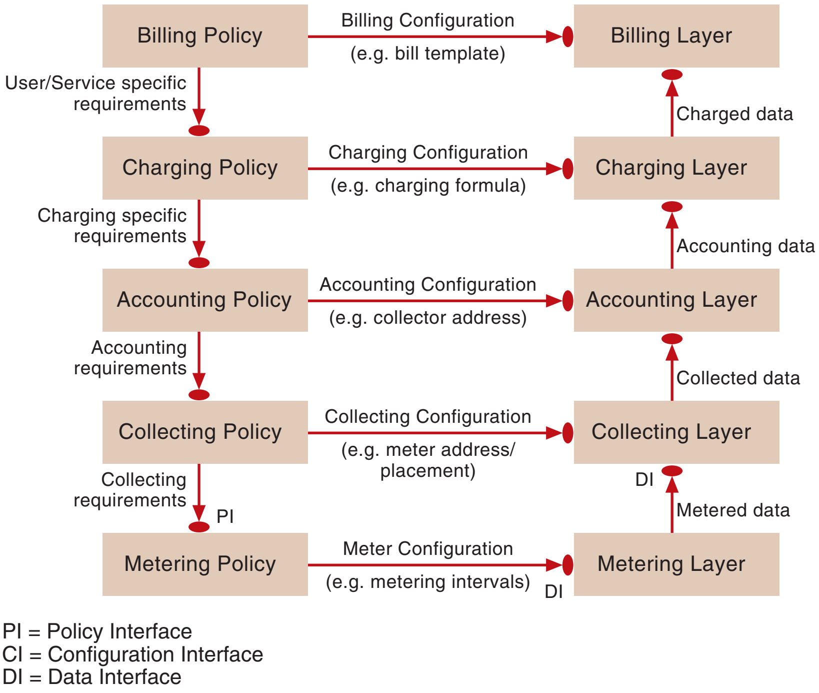
6 QUASI-model and Charging

A number of processes are involved from capturing the usage to creating a bill to be sent to a customer. A layered model can represent these processes. The charging and accounting reference model used in P906-GI is illustrated in Figure 7.

Each layer represents a specific basic functionality, which is configurable by using parameters supplied by specific policy definitions.

The **metering layer** tracks and records the usage of resources by observing the traffic flows. The metering policy, used for configuring the metering layer, specifies the attributes of the traffic flows to be observed. In a connectionless net-

Figure 7 A Charging and Accounting Model used in P906



work, such as Internet, where it is difficult to locate when a flow is finished, the metering policy can also be used to define the flow duration.

The **collecting layer** accesses data provided by metering entities as well as collecting charging related events and forwarding them for further processing to the accounting layer. This layer can collect information from multiple meters, as for multicast, and distribute to home domains, as for user roaming. For this reason, the efforts in standardising data exchange format and protocol at this layer will be beneficial. The meters from where to collect the data, the type of data and the frequency in collecting them are defined in the accounting policy.

The **accounting layer** consolidates the collected information from the collecting layer either within the same provider domain or from other provider domains and creates *accounting data sets* or *records* which are passed further to the charging layer. For supporting multicast charging, the multicast topology including splitting points can be reconstructed by entities of this layer.

The **charging layer** derives charges from the accounting records based on service specific charging and pricing schemes, which are specified by the charging policy. This layer basically translates technical values (i.e. measured resource reservation and consumption) into monetary units using a charging formula. As a result of this process, a charging record is created.

The **billing layer** collects the charging information (given in charging record) for a customer over a time period, e.g. one month, and includes subscription charges and possible discounts into a bill. Billing policy can be used to specify the bill details.

Naturally, when building a particular charging and accounting system, not all components have to be included. For example, a service provider who provides only one service and charges the customers on the flat-rate basis can implement only the functionality of the billing layer. On the other hand, a service provider offering multiple services may implement the policy-based architecture to allow different charging schemes to be used for different services or customers without having to hard-code the charging formula into the billing system.

The accounting architecture is based on the charging and accounting reference model. It consists of different processes that take over the functions for the different layers of the reference model. The processes are controlled by policies which provide configuration information in accordance with the needed accounting task. The policies describe the flows or traffic aggregates that should be measured, the attributes that have to be stored, the collection intervals, data aggregation instructions, etc. With this flexibility with regard to charging schemes, used QoS provisioning technique, traffic mix and user profiles can be achieved. With the usage of standardized policies it is also possible to instruct different types of the accounting components (e.g. different meter processes) in a consistent manner. With this, different infrastructures that might be used in different administrative domains can be supported.

Two different architectural approaches built on this model have implemented the QUASI-model, one centralised approach (developed by Deutsche Telekom T-Nova in collaboration with GMD Fokus) and the other distributed approach (developed by BT Labs in collaboration with

Notation used for charging schemes

Charging scheme parameters:

Access_charge: Part of the total charge for network services that can contain a one-time site connection fee and monthly rental.

Usage_charge: Part of the total charge for network services that is associated with resource reservation and consumption.

$\boldsymbol{x}_i$: In the case of transport services, includes the Network Performance Level (NPL) and traffic profile contained in the Quality Class Specification (QCS) for service *i*. In the case of end services, includes the service (application) and quality class.

$p_T(\boldsymbol{x}_i)$: Charge per unit of time (e.g. per minutes).

$p_V(\boldsymbol{x}_i)$: Charge per unit of volume (e.g. per Mbyte).

$p_c(\boldsymbol{x}_i)$: Per connection charge (applies only to connection oriented services).

T^i: Duration of service or connection *i*.

V^i: Transferred volume during service or connection *i*.

Compensation scheme parameters:

$p_{T,reduced}$: Reduced per unit of time charge, when corresponding NPL is violated.

$p_{V,reduced}$: Reduced per unit of volume charge, when corresponding NPL is violated.

Vnc^i: Volume of non-conforming traffic for service *i*.

Tvg^i: Duration of period in which NPL is violated for service *i*.

Vvg^i: Volume transferred during period in which NPL is violated for service i. This variable can either be measured directly by the QoS monitoring system, or estimated from other measurements.

Vvg: Volume (of all flows corresponding to a given NPL) transferred during period in which NPL is violated.

UCL (University College London). The implemented solutions have then been analysed through experimental tests. More detail on these can be found in [P906-7], [P906-8].

7 Charging Schemes

After considering the charging and accounting model and architecture, the charging of both connectivity/transport and end-user services is discussed in this chapter. The simplest usage-based charging scheme, which can be applied to both connectivity/transport and end-user services, and which considers volume in addition to time as a measure of usage, is the scheme where the charge is calculated by using a simple function that is linear in measurements of time and volume. The parameters of this function, which represent a charge per unit of time and a charge per unit of volume, are a function of the parameters of the Quality Class Specification (QCS), namely the NPL, which contains a set of target values (or range of values) for relevant Network Performance Parameters (NPP) that are guaranteed by the network, and the traffic profile, which describes the maximum amount of conforming traffic that the user can send. This will be discussed in more detail later. The notation used in the following subsections is given in ingress A.

A *charging scheme* is an algorithm for calculating the charge for some network service[2]. In the case of telecommunication services, a user's charge is calculated based on accounting data that contain information regarding the resource consumption for that user, and prices from tariff tables published by the provider. Bearing in mind the business model described in Section 2, network connectivity services are offered over the SP-NP interface. Such services involve the transfer of bits with, possibly, some performance guarantees, but without any knowledge of the higher layer application that generated the traffic. End-services are those offered over the user – SP interface. Charges for such services include both connectivity/transport level charges and application level charges, where the latter depend on the particular service (application) offered. Basically, charging for network transport service involves charges for a basic service, while charging for end-services involves both charges for the basic services and for value-added services.

In general, a charge for network connectivity services may include a subscription[3] component and a usage component. The *subscription component* can consist of a one-time site connection fee, which is paid once when the user's network is connected to the provider and corresponds to the cost of equipment and labour necessary for connecting a customer's network with the provider, and a rental (e.g. monthly) that is associated with facilities in the access portion (e.g. router ports), as well as operational and maintenance costs. The *usage component* is associated with resource reservation and consumption in the backbone. It can consist of the following:

- A fee for setting-up a connection (for connection-oriented services like IP telephony), that is associated with the signalling required to set-up the connection and the maintenance of related state information.
- A charge for consumed resources, which depends on measures of resource usage such as the duration (time), the volume transferred, quality class. Such measures are captured by charging and accounting systems and are included in the accounting data.

Considering all parts mentioned, a customer's charge for network connectivity services can be expressed as follows:

$$\begin{aligned}&Charge = \\ &Access\_charge + \sum_i Usage\_charge_i \end{aligned} \tag{1}$$

where $Usage\_charge_i$ is the charge for connection i. In the simple case where duration (time) and volume are the only measures of resource consumption, then the usage charge can be expressed as follows:

$$\begin{aligned}&Usage\_charge_i = \\ &p_T(\boldsymbol{x}_i)T^i + p_V(\boldsymbol{x}_i)V^i + p_c(\boldsymbol{x}_i)\end{aligned} \tag{2}$$

where $\boldsymbol{x}_i$ describes, in the terminology of the QUASI-model, the NPL and the traffic profile for connection i. The measured variables T^i and V^i are the duration and transferred volume, respectively, for connection i. Finally, the prices p_T, p_V, and p_c are the price per unit of time (e.g. per minute), the price per unit of volume (e.g. per Mbyte), and the connection set-up fee (e.g. per connection), respectively. The price p_T corresponds to the amount of resources reserved, whereas the price p_V corresponds to the actual amount of resources used. In addition to technical considerations, expressed through the dependence on $\boldsymbol{x}_i$, these prices will depend on eco-

[2] *In this section, the term "network service" is used to refer to either end-user service or network transport service.*

[3] *Note that this component is sometimes in literature referred to as the access fee/component.*

nomic issues such as market structure and demand.

Equation (2) is rather general and can describe a wide range of the charging schemes that are present in today's telecommunications market, for both guaranteed services and best-effort services. Equation (1) can be further generalised by including time-of-day pricing, which allows prices to depend on the specific time-of-day the service is delivered. The rationale for such a charging scheme is to provide incentives for users to move traffic which they value less to off-peak hours (for which the prices are lower compared to on-peak hours), hence achieving a more uniform utilisation of network resources throughout the day.

7.1 Charging for Network Connectivity Services

In addition to recovering the costs for service provisioning and generating revenue, charging may play an important role for controlling resource usage. When demand is always less than supply, the controlling function of charging is not so important. On the contrary, if demand exceeds supply, charging can be used as an effective mechanism for controlling how resources are used, and help the network achieve efficient and stable operation. In such cases, in order to provide incentives for users to use the network according to their actual needs and to charge them in a fair way, charges need to take some account of resource usage. Furthermore, by setting prices appropriately, usage-based charging can generate the amount of revenue necessary for expanding the network to meet the excess demand.

The issues of measurement methods for the resource usage in networks supporting bursty traffic and different ways of constructing charging schemes using these measurements, are discussed next. Note that the discussion refers to the usage component of a user's charge, which is associated with resource consumption. The schemes discussed here are appropriate for charging wholesale transport services, e.g. a service provided by NP to SP.

7.2 Measuring Resource Usage

The amount of resources used by a user generating bursty traffic can depend on the user's traffic profile, the NPL guaranteed by the network, and the statistical characteristics of the user's traffic. It is desirable to map such multi-dimensional quantities into a scalar that reflects the relative amount of resources used by the user. This scalar is typically called the "effective rate" or "effective bandwidth" of the user's traffic stream, and can simplify the problem of charging a network service based on the relative amount of resources used by the service.

In the case of best-effort services, the effective bandwidth of a stream reduces to its mean rate. This can be understood as follows: For best-effort services, there are no performance guarantees. The only requirement is that traffic eventually reaches its destination. Considering a single link, the latter requirement translates to a stability condition for the link that can be written as

$$\sum m_i \leq C ,$$

where m_i is the mean rate of a traffic stream i and C is the link capacity. Hence, the mean rate is the appropriate measure of resource usage for best-effort services.

In the case of guaranteed services, the actual effective bandwidth of a traffic stream is a complex function, and one usually considers bounds of the actual effective bandwidth that are simpler and involve easy to measure quantities. In the case where the only measure of resource usage is the mean rate, the bound can be written as $B(\boldsymbol{x},m)$, where $\boldsymbol{x}$ includes the NPL and traffic profile, and m is the mean rate of stream. It can be shown that this bound is a concave function of the mean rate m [CA$hMAN].

The concavity of the effective bandwidth bound is large when the peak rate is high relative to the network capacity or when the QoS guarantees are tight (e.g. small delay and small loss probability). This concavity property can be used to provide interesting incentives to the users. On the other hand, for best-effort services the curve becomes linear, i.e. the effective bandwidth is equal to the mean rate of the stream, as discussed above.

More complex bounds that depend on more detailed statistics than the mean rate would result in higher accuracy. However, investigations and trials have shown that higher complexity, hence greater difficulty to understand charging schemes based on such bounds, can easily outweigh the advantage of higher accuracy [CA$hMAN].

7.3 Charging for Resource Usage

The approaches, discussed before, for measuring resource usage provide input to the charging scheme. For best-effort services, it was mentioned that the appropriate measure of resource usage is the streams mean rate. Hence, the charge per unit of time is $p(\boldsymbol{x})m$, with $\boldsymbol{x} = \{\text{best-effort}\}$. The parameter $p(\boldsymbol{x})$ is the price per unit of rate and unit of time for best effort services; this price will typically depend on economic factors such as demand and competition. Hence, the total usage charge for best-effort services is given by:

$$Usage\_charge_{best\text{-}effort} = p_V(\boldsymbol{x})V$$

where $\boldsymbol{x}$ = {best-effort} and V is the transferred volume.

For services with QoS guarantees, a straightforward approach would be to set the charge per unit of time equal to $pB(\boldsymbol{x}, M)$, where $B(\boldsymbol{x},M)$ is the effective bandwidth for contract $\boldsymbol{x}$ and mean rate M, and p is the price per unit of effective bandwidth; the latter price is determined by economic factors such as demand and competition). In this case, the usage charge is then

$$pB\left(\boldsymbol{x}, \frac{V}{T}\right)T ,$$

where T is the duration and V is the transferred volume. A disadvantage of such an approach is that charges are not linear functions of the measurements of duration and volume, thus making it difficult for users to understand.

Interestingly enough [CA$hMAN, SoKe97], based on the effective bandwidth bound as a measure of resource usage, one can construct a charging scheme linear in measurements of duration and volume that approximate the previous charge. This charging scheme is presented to the users as a trade-off between a duration charge and a volume charge. In particular, given his NPL and traffic profile, the user is offered a set of charging parameters ($p_T(\boldsymbol{x})$, $p_V(\boldsymbol{x})$) to choose from. The parameters $p_T(\boldsymbol{x})$, $p_V(\boldsymbol{x})$ represent a duration and a volume charge, respectively. The usage charge will be

$$Usage\_charge = p_T(\boldsymbol{x})T + p_V(\boldsymbol{x})V$$

In practice, for a given SLA, the provider can offer a small number of tariff pairs.

A number of additions/modifications can be made to the basic approach described above. If the service is connection oriented, then one can include in the usage charge a connection set-up fee p_c [SoKe97]. This fee accounts for the signalling resources required to set up the connection and the state that needs to be maintained throughout the duration of the connection. In addition, a discount for higher volume connections can be incorporated in the scheme.

In the case a user generates and injects the traffic that is not conformant (exceeds the conditions) with the traffic profile agreed in the SLA between the user and the provider, various reactions can be initiated. One such reaction is to mark such traffic for dropping, and charge it at the same rate as best-effort.

7.4 Charging for End Services

Charges for end-services can be given by a formula similar to Equation 1, namely:

$$Service\_charge_i = p_T(\boldsymbol{x}_i)T^i + p_V(\boldsymbol{x}_i)V^i + p_c(\boldsymbol{x}_i) \quad (3)$$

where now the parameter $\boldsymbol{x}$ denotes the service (application) and quality class selected by the user. These parameters are included in the SLA between the user and the SP.

A charge for end-service may include the charge for the corresponding connectivity necessary to provide the service, and charges for the service itself. The latter can include content charges in the case of e.g. video delivery.

Recall the business model described in Chapter 2, where the SP offers a different quality class to the user according to the applications he might use. Hence, the SP "hides" from end users the low level details of the QoS parameters and their corresponding values. In the same sense, an SP may wish to provide very simple tariffs, even if they lose some, or even all, the structural characteristics of the transport level charges. The motivation for that might be to attract the customers by having a very simple charging scheme, e.g. flat-rate charging, or because the transport level charges are a small percentage of the charges for the service itself (e.g. the content charge). Indeed, economic and marketing issues may have a significant effect on both the structure of the charging scheme for end services and the corresponding prices.

As mentioned before, transport level charges at the SP-NP interface will influence service level charges. Consider the following examples:

- For an SP offering a video playback service, the characteristics and requirements (e.g. in terms of bandwidth) for a particular video are known in advance. Indeed, these requirements depend not only on the content but also on the resolution of the video encoding. Knowledge of the requirements will in turn enable an SP to estimate the corresponding charges of the transport level services required by the network provider for the delivery of the particular video. Of course, different video streams may have different transport level requirements. The service provider, however, may select to offer simple duration-based charges with the same prices for all video streams which, when averaged over all connections of many users, will absorb the varying transport level charges.

- For an SP offering IP telephony or videoconferencing services, the characteristics and requirements for a particular session are not known in advance. However, the SP can have empirical measurements of past sessions, hence can determine the average transport level requirements of one session. Furthermore, the estimates of transport level requirements can be refined with time.

In the above service examples, the SP might choose to charge based on duration only. For other services, however, the SP may select to charge also based on the transferred volume, if it sees that providing users the incentive to control the total volume they transfer is important. One example of such a service is Internet access over ADSL. In this example, if the SP does not provide incentives for users to limit their transferred volume, then congestion in some part of the access network can arise.

7.5 Compensation Schemes

By agreeing the SLA, both the customer and the provider define the behaviour, duties and rights of each of the parties. Therefore, in case some of the statements in the SLA are not fulfilled, a reaction pattern can be applied. Note that the reaction pattern is described in the SLA as well. One reaction is related to the compensation schemes. Such schemes imply the compensation the provider offers to the user after not fulfilling the conditions given in the SLA (and under condition that the user's traffic was conformant with the agreed traffic pattern).

In a QUASIMODO scope, a user should be charged as agreed in the SLA, but only if the agreed NPL is delivered as well. In case the provider failed to deliver the NPL with the guarantees stated in the SLA, some compensation may be invoiced. Note that the compensation is not restricted only to the direct money flow, but can impose "service credit", i.e. service usage free of charge for a certain (defined) period. When constructing compensation schemes, the NPL violations are necessary information. Hence, measurements related to this dictate the granularity a compensation scheme can be developed with. Depending on the type of measurement/monitoring methods and tools, the detection of the violation of a certain QoS parameter will differ, and hence affect the compensation scheme structure. For example, whether detection of NPL violations is for individual flows or for aggregate flows, whether the specific NPL parameters that are violated are detected, whether the duration of NPL violations and the traffic transferred during these violations is measured.[4]

More details on the compensation schemes discussed in the QUASIMODO project can be found in [P906-7], [P906-8]

8 Open Issues and Concluding Remarks

The question of scalability of the QUASI-model deserves further attention. One particular problem is scalability of a central management system for one operator's network. If NPL measurements are made with test traffic so that traffic is sent from one MRP, loops back from another MRP, and the measure for the NPL-parameters is derived as an average from the two-way delays and losses, then one MPR can do its measurements alone. This was applied in the QUASI-model DiffServ implementations. In the QUASI-IntServ implementation delays are measured one-way and clock synchronisation is needed, but no correlation of traces of measured times of packet arrivals in two MRPs is needed, as the delay is calculated from a time stamp. In both implementations measurements from MRPs are collected to a central point, either continuously or if measured values are unnecessarily good or too bad. The same or another central point also manages dynamically the MRPs in the QUASI-IntServ implementation. A central point may be considered a poorly scalable solution. It is possible to divide the network to subnetworks and set target values to each subnetwork and to have a hierarchical management system where a central manager manages subnetwork managers, in the same way as in distributed network management of SNMPv2/3. The QUASI IntServ implementation uses a simple unsecured socket connection between the managing node and the MRPs. A suitable de-facto standard, like SNMP or GSMP (General Switch Management Protocol) would be an improvement, but the protocol should be secured as changing bandwidths of the RSVP flows between MRPs make the network vulnerable to denial of service attacks.

An important issue is security. If security is provided with IPSec (or IPv6 security), then IPSec restricts the use of NAT in the QUASI-IntServ and if EPS with data encryption is used, IPSec prevents reading the port and the protocol, also AH prevents NAT. Though the TOS field can be updated, transport mode IPSec with EPS and encryption conflicts with the use of DiffServ also. A working solution is to use IPSec in the tunneling mode and terminate tunnels to MPRs and continue from there with another tunnel. If the network is divided into subnetworks because of more scalable management, it could induce long delays to terminate IPSec tunnels to each

[4] *An issue we do not discuss here is which system (accounting or QoS monitoring) is responsible for collecting such measurements.*

intermediate MRP. This is fortunately not needed as the IP-packets are already marked for the desired NPL and it is sufficient to read the outer IP-header. In this way the QUASI-model can be used with decentralised management and IPSec. Offering VoIP with IPSec tunnels may cause problems because of header overhead and set-up delays connected with IKE, but these problems are not specific to the use of IPSec in the QUASI-model.

Currently the implementations for measurement and management do not have security mechanisms in the user access. A full AAA-protocol (Authentication, Authorisation, Accounting) could be used. The present AAA protocols, like COPS, RADIUS or Diameter, are not ideal for the QUASI-IntServ. The implementations for charging the use of AAA, see [P906-6].

Another scalability issue is the usage of access lists in the DiffServ implementation. The problem appears because a user can select different quality classes for an application in the QUASI-model. In conversational services, like VoIP, a subscriber to better quality should receive better quality in a conversation with a user of worse quality. This is solved in the DiffServ implementation by putting all subscribers to better quality to access lists of all MRPs. Then the MRP knows to mark the TOS field of IP packets destined to a subscriber to better quality, but the solution is not scalable. One way could be to upgrade SIP or H.323 so that the applications negotiate the used quality. This solution may be difficult to enforce as SIP and H.323 compliant implementations exist and the standards do not yet include quality negotiation, and requiring upgrading applications to support the QUASI-model before the QUASI-model can be used hinders its usage. In the QUASI IntServ the problem is solved by MRPs keeping the state of existing connections and exchanging the information of AC/QC in both MRPs of a connection. A similar solution might be the easiest way to scale the DiffServ implementation.

The QUASI-model does not restrict the use of MPLS inside the network, but it should be noted that in the QUASI-model an application can be assigned a different quality. In many label switching methods traffic flows are classified by their traffic pattern and the classifier decides what is the required quality of the flow. This approach fails in the QUASI-model as the traffic pattern does not indicate what the desired quality is.

Mobility via Mobile IP (or IPv6 mobility) has not been considered in the QUASI-model. It is likely to require changes to the QUASI-model. In Mobile IP traffic is sent to the home agent and then forwarded to the visiting agent. In order to support the same quality in both connections, it seems that the MRPs should have signalling to inform the MRP on the route to the visiting agent what is the selected AC/QC.

The scalability problems of the DiffServ implementation can be solved with additional protocols and enhancements of the model. They add some complexity but also they add some functionality. Service differentiation is basically a philosophical matter: should one worsen existing quality simply to provide service differentiation? Does it mean provisioning guaranteed bad service? The solution we proposed to be used in QUASI-IntServ is connection blocking, but this implies some kind of connection oriented view to a traditionally connectionless IP-network. In general, is QUASI-IntServ a poor version of ATM and why not to use the real ATM? Anything based on IntServ and in conformance with the QUASI-model is likely to be similar to QUASI-IntServ.

There are proposals to use charging schemes where the operator increases the prices when the network is heavily loaded and the users are expected to respond with reduced traffic demand. In the situation when the users are connected to a single operator and they respond to prices, these methods work as feedback methods and the only stability issue is the control loop delay. In the QUASI-model there is an intermediate role – the retailer – and the retailer can be connected to several operators, some of which may use congestion pricing. The retailer, let us call it the ISP, may try to shift traffic to an operator offering lower prices instead of reflecting the increased prices to end users as long as there are cheap operators available. Then the users do not respond to the change or prices, as their prices do not change, and the total traffic demand is not reduced. In the QUASI-model traffic control is on the IP layer and connectionless, therefore existing connections can be shifted on-the-fly to another operator providing sufficient NPL. In this situation the response to congestion prices is On/Off, i.e. the ISP minimising costs for each charging time slot will shift traffic abruptly from a more expensive operator to a cheaper one. It is possible to use more complicated operator pricing schemes, like carry the first N-bytes in a charging time slot with come cost per byte, and the remaining traffic on the time slot is more expensive. Then the operator will get the response it desires in congestion pricing schemes. This shows that the role of the retailer is useful: the pricing scheme of the operator is unfair as prices depend on the arrival times of IP packets with respect to charging time slots, but the ISP can offer users fair and more constant prices.

There are other similar observations in a multi-operator situation. If there is an operator offering predictable prices, while some other offer load or volume dependent prices, there may be a protection strategy for the ISP. This means that it can offer fair prices (in the sense of a martingale measure) to end-users and have the strategy of shifting the traffic to the predicable prices in case lower usage-based prices are not offered by other operators.

Another observation is that there are cases when changes in traffic volume are predictable, like the traffic variation during the day to a large extent is. If volume changes are predictable and there are operators offering flat prices and others offering usage based prices, the ISP may shift traffic on high traffic times to the flat rate and on low traffic times to the usage based rate. These may influence the viability of a particular pricing scheme, and there are also countermeasures. In this example the operator may require that capacity is bought only for time units of a certain duration, that predictability of traffic cannot be used by an ISP for unwanted optimising purposes.

This paper described some results of the QUASIMODO-project, with emphasis on the parts where the authors of the paper worked. The QUASI-model implementations for measurement and management contain measurements for QoS monitoring, mapping users' (QC, AC) selections to DiffServ and IntServ architectures, and some additional QoS management methods. The charging-related work in the QUASIMODO-project included both generic charging and accounting model, as well as two possible billing system solutions. Here, the theoretical findings have been stressed, while more details on the particular implementations can be found in [P906-7], [P906-8]. The linking between the measurement and management implementations and the billing systems was not demonstrated; it is realised by the possibility of obtaining NPL-measurements to the charging system.

Acknowledgement

Although this paper represents the view of the authors, this way we would like to thank all the P906-GI participants for the interesting work and discussions undertaken. A special thanks goes to Mr. Imad Ossaily and Mr. Denis Karpov, who worked with the QUASI IntServ implementation. Several figures are from the QUASIMODO deliverables.

11 References

[Alt00] Altman, J, Rupp, B, Varaiya, P. 2000. Quality Matters: Some remarks on Internet Service Provisioning and Tariff design. *Telektronikk*, 96 (2), 20–25.

[CA$hMAN] Songhurst, D J (ed). 1999. *Charging Communication Networks: From Theory to Practice.* Amsterdam, Elsevier. ISBN 0-444-50275-0.

[P906-1] *EURESCOM P906-QUASIMODO: Deliverable 1 – Offering Quality Classes to end users.* Heidelberg, EURESCOM, May 2000.

[P906-2] *EURESCOM P906-QUASIMODO: Project Report 2 – Methodologies and tools for QoS measurement and management.* Heidelberg, EURESCOM, December 2000.

[P906-3] *EURESCOM P906-QUASIMODO Technical Information 1 – Survey of existing methodologies and tools for QoS measurement and management.* Heidelberg, EURESCOM, December 2000.

[P906-4] *EURESCOM P906-QUASIMODO Technical Information 2: QUASI-model Implementation.* Heidelberg, EURESCOM, December 2000.

[P906-5] *EURESCOM P906-QUASIMODO Technical Information 3: Experimental evaluation of QoS measurement and management.* Heidelberg, EURESCOM, December 2000.

[P906-6] *EURESCOM P906-QUASIMODO Technical Information 4 – Key QoS charging issues.* Heidelberg, EURESCOM, 2001.

[P906-7] *EURESCOM P906-QUASIMODO Deliverable 3 – Methodologies and policies for QoS charging.* Heidelberg, EURESCOM, February, 2001.

[P906-8] *EURESCOM P906-QUASIMODO Technical Information 6 – Summary of QUASIMODO Findings on QoS.* EURESCOM, Heidelberg, February 2001.

[SoKe97] Songhurst, D, Kelly, F. 1997. Charging Schemes for Multiservice Networks. In: *Proc. ITC-15.* Ramaswami, V, Wirth, P E (eds). Amsterdam, Elsevier.

http://www.eurescom.de/Public/Projects/p900-series/P906/P906.htm

Network Principles and Applications

TERJE JENSEN

Terje Jensen (39) is Research Manager at Telenor R&D, Kjeller. He earned his PhD degree in 1995 from the Norwegian University of Science and Technology. Activities include performance modelling and analysis, dimensioning and network evolution studies.

terje.jensen1@telenor.com

Although the IP-based networks are said to have inherent features that differ from the "traditional" telecommunication networks, there are quite a lot of similarities. This paper points out the similarities by elaborating on generic capabilities being present in all wide area networks. However, the IP "specifics" are also included. These are more directed on routing and the use of IP in relation with other protocols and systems. In particular, relations with optics, mobility and multicasting are discussed. The Web and support of the Virtual Private Network and telephony services are also described.

1 Introduction

It is confirmed by several sources that the growth in the Internet has been phenomenal. This is counted both in volume of traffic and number of connected users and hosts/sites. The Internet together with the mobile services may not see any cases of comparable growth rate in the industrial countries. However, not to forget that telephony networks are also intensively deployed in several parts of the world. However, the Internet Protocol (IP) is entering into both mobile networks and voice networks. This is for example observed by IP being the current longer-term solution for the 3rd generation mobile systems and work on finding suitable configurations for providing telephony over IP.

It may therefore be claimed that IP works as a common "glue" between the applications and the underlying infrastructure, see Figure 1. Applications are the functions towards the end-user, such as a human being. Infrastructure is the underlying functions such as cabling.

However, it should be observed that much work is being carried out to find efficient configurations and mechanisms built around IP. This can be referred to as IP++, including the portfolio of functionality promoted, e.g. for routing, resource handling, traffic control, multicast, and so forth.

Looking at several major operators one sees that a consolidation of networks is strived for. A portfolio of any larger operator includes PSTN/ ISDN, X.21 networks, X. 25 networks, ATM network, Frame Relay networks, and others. Arriving at fewer networks is considered a better configuration, taking into account the economy of scale and scope. However, it also raises several challenges, preserving the mechanisms for differentiating between traffic flows, e.g. according to their characteristics, level of payments of the customers.

Tuning IP++ to work as a common layer for a range of applications is not a straightforward task. Therefore, several projects related to multiservice IP-based networks have been initiated. In addition to the technical challenges, there are also others for example in the area of business modelling, regulatory, property rights, etc. The latter are not considered to any extent in this issue of *Telektronikk*, but should not go unnoticed.

Within certain limits, one may say that arriving at more predictable services is one of the main goals of the IP-related work. This is also addressed by the IP++, and addressed in accompanying papers of this *Telektronikk* issue. Predictability is not to be understood in a restricted

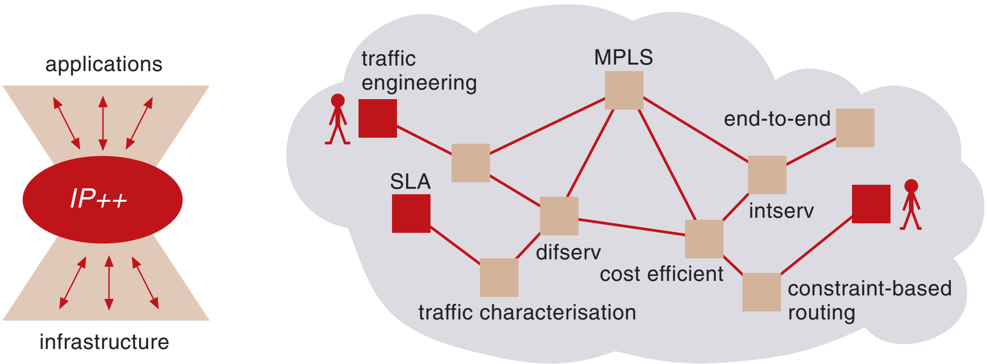


Figure 1 IP ++ as the common "glue" for carrying traffic flows from different applications over different infrastructure types

sense. In fact, traffic from most of the traditional applications on IP-based networks is quite tolerant with respect to variations in transfer times (also referred to as elastic traffic flows). However, too wide changes in the behaviour (such as longer response time) would likely be considered a great annoyance for a human user.

The main topics addressed in this article are related to IP and how it can be utilised. Accompanying articles consider the aspects more related to Traffic Engineering (TE). Concepts on domains and layers are described in the following chapter. Chapter 3 addresses selected topics on optics from an “IP point of view”. Mobility and multicast are discussed in Chapter 4 and Chapter 5, respectively. Authentication and other security issues are treated in Chapter 6. A few scenarios and examples, including Virtual Private Network, Web browsing and telephony are given in Chapter 7. The main objective of this article is to provide a soft introduction to the application of IP and some core functions that have to be present in a wide area network.

2 Network Components

2.1 Concepts

Being no surprise to anyone, a network is put together by a number of components; each of them having its characteristics. Moreover, some of the components can be grouped based on certain characteristics. This is commonly useful when discussing appropriate solutions for the components and how they can be put together to compose a network. That is, each of the components would implement certain mechanisms, be designed according to an architecture, and so forth.

Looking at studies deriving systematic network descriptions, one frequently observes a division into domains and strata, see Figure 2. *Domain* refers to geographical separation, while *stratum* refers to a functional separation. These are depicted by horizontal and vertical separations. A basic example of a set of strata is the 7-layer Open System Interconnection (OSI) model. Here, however, stratum is used in a more general interpretation, although similarities with the OSI model can be recognised.

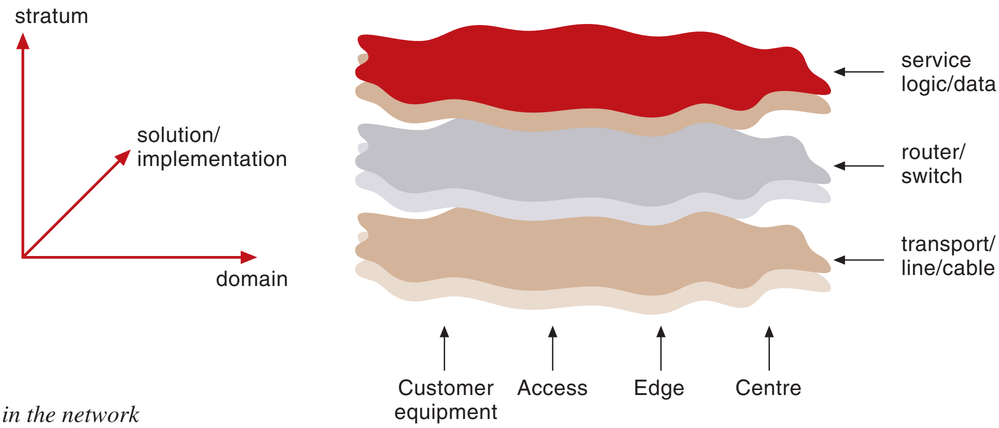


Figure 2 Identifying components in the network

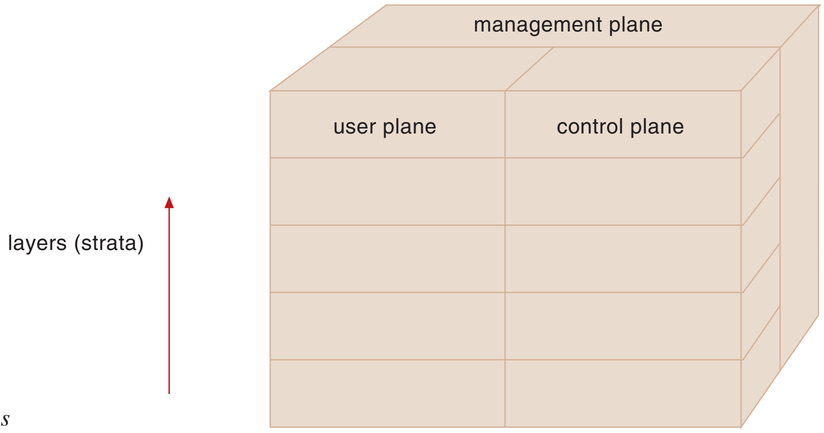


Figure 3 User, control and management activities

Zooming in on IP-related aspects, the stratum referring to routers is commonly placed in the centre. Several of the relations with upper and lower strata (referred to as layers) impact the behaviour of IP. So, when discussing interconnecting IP routers (and other IP elements) one should keep in mind that lower layer functionality is used for carrying the IP packets. IP packets are used for carrying higher layer information, and other functions are used for finding where to direct the information. According to OSI, IP will be placed in layer 3, while lower layers are 2 and below and upper layers from 4 and above.

Besides carrying information from higher layers, other types of functionality are also present in a network. Typical classes are control and management. The actual distinction between user data, control and management might be rather blurred in some cases. However, schematically this can be illustrated as in Figure 3. Some refers to this as the cube of the International Telecommunication Union – Telecommunication Standards Sector (ITU-T).

Briefly, the main purposes of each of the planes are:

- *User plane* – to convey the user information (information from higher layers);
- *Control plane* – to control traffic flows and resource configurations;
- *Management plane* – to manage network resources, including fault management, configuration management, accounting management, performance management, security management.

Any larger network would typically have functions belonging to all these planes, although the way functions are implemented varies. An objective is to find efficient complete solutions and combinations of functions that incorporate all needed functionality.

In order to locate where to direct the information, addresses and routing functionality have to be present. So, addresses are used for identifying a unit/interface, while routing is used to find how to direct the information towards the address (and the unit/interface it represents).

2.2 Domains

Domains are commonly identified according to some kind of geographical arrangements. That is, separate domains are physically dispersed, and likely to be interconnected at certain points. When traffic has to traverse a number of domains, this can be schematically depicted as a chain, see Figure 4. Interconnection points are realised at *border* nodes, e.g. at a border router. When interconnecting a customer to the core network, a border router is frequently called an *edge* router.

The expression *Autonomous System* (AS) is frequently seen. A common understanding of an AS is a set of routers under a single technical administration. This implies that an AS has its characteristics and is commonly managed by a single organisation (within the same trusted unit).

Examples of domains are the customer premises network (e.g. Local Area Network, LAN, and terminals), access network and core network. These may well have distinct characteristics implying that which solutions to use for each of them differ. For example, the limited coverage of a LAN allows for adding transmission capacity without radically increasing the cost. Moreover, a company would often own its LAN such that sophisticated mechanisms for accounting/charging may not be needed. The same goes for security solutions, although certain protection means are typically introduced, like passwords, limited access rights, and so forth.

An access line may be dedicated to a single customer, resulting in relatively low average utilisation (e.g. for a residential customer). Hence, commonly a significant fraction of the overall cost is associated with the access network. Shortening the dedicated capacity introducing multiplexing/concentrating equipment is often used seeking to reduce the overall cost, see Figure 5. For several networks the relative cost of the access portion is fairly high, commonly in the area of 60 % – 80 % for public networks, although this depends on the solutions used.

On the IP level the access network commonly looks like a tree or a star network, with the edge router located at the root (or in the centre). However, on the transmission layer ring structures could be used, e.g. for dependability arguments.

Looking at the access link for a single user, the capacity of the links should be according to the maximum demand from that user. However, the peak load (bit rate) may very well be rather high compared to the average load, e.g. during the day. An analogy is found in the telephony net-

Figure 4 Schematic illustration of interconnected domains

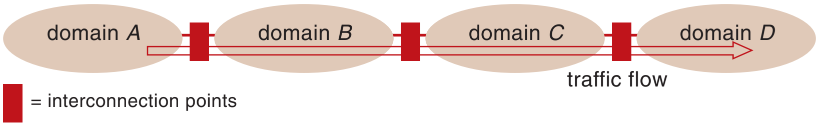
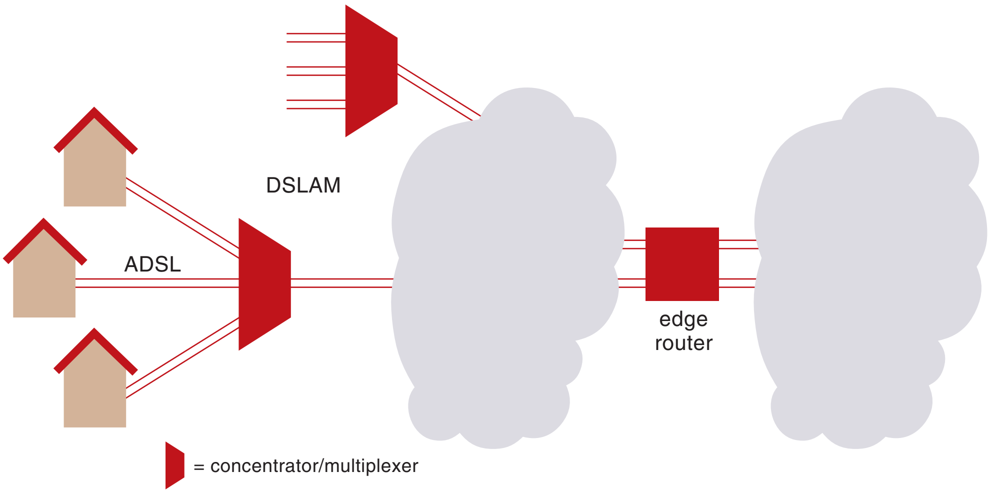


Figure 5 Introducing equipment for concentration and multiplexer to reduce the cost for dedicated access lines per user



work were the access link is in use during a phone conversation, but idle most of the time. Installing a capacity much higher than the average load is a reason for the rather high cost of the access network.

As many types of applications and users will be connected, a range of access capacities could be requested. Commonly, however, rather fixed capacities are used, e.g. offering Asymmetric Digital Subscriber Line (ADSL) downstream rates as 384, 768 and 1024 kbit/s. This also contributes to the potential capacity of the access network usually being poorly utilised.

A core domain usually carries traffic from a greater set of customers resulting in higher capacity links and routers. A certain averaging effect of the traffic, e.g. during the day is also commonly observed, for example related to the different customer types and use of services.

The edge of the core network usually has a number of features related to individual users, like access lists and monitoring mechanisms. Edge routers will then implement such features. Towards other operators, border routers, possibly with similar functionality can be present.

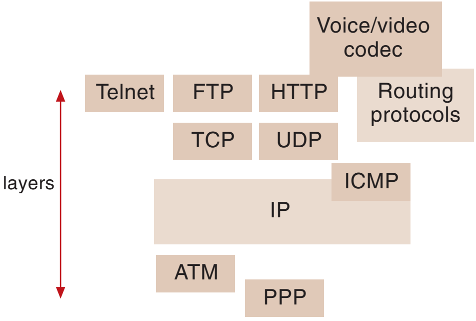


Figure 6 Layers and example of protocols related to IP

Providing efficient traffic handling implies that appropriate solutions must be available in all the domains involved. However, for customer equipment, it is usually expected that the cost of providing capacity is relatively low, implying that capacity is added in case a performance problem emerges. For other domains mechanisms for differentiated handling of services (and traffic flows) seem to be gradually introduced. In addition to differentiating between services, differentiating between customers would also be likely (although this could again be seen as a kind of service differentiation).

When looking for potential performance bottlenecks it is essential to include the end systems as well as the processing capabilities in all domains. This means that efficiency of protocol software (as well as software of other traffic handling mechanisms) is a pivotal point. As the capacity evolution of transmission outpaces the computer capacity growth, the system designers may rather concentrate on achieving high transfer speed (putting data on the output interface) than optimised utilisation of the transmission bandwidth when seen from an end system point of view. The network operator view may be more balanced, however, as a huge number of traffic flows will be present.

2.3 Layering

Similar to the OSI model (Open System Interconnection), the Internet-related protocols can also be depicted as a hierarchy, see Figure 6.

Numerous protocols could be used below IP, like Asynchronous Transfer Mode (ATM) and Point-to-Point Protocol (PPP). A number of protocols could also be used in the transport layer, although Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are amongst the most popular ones.

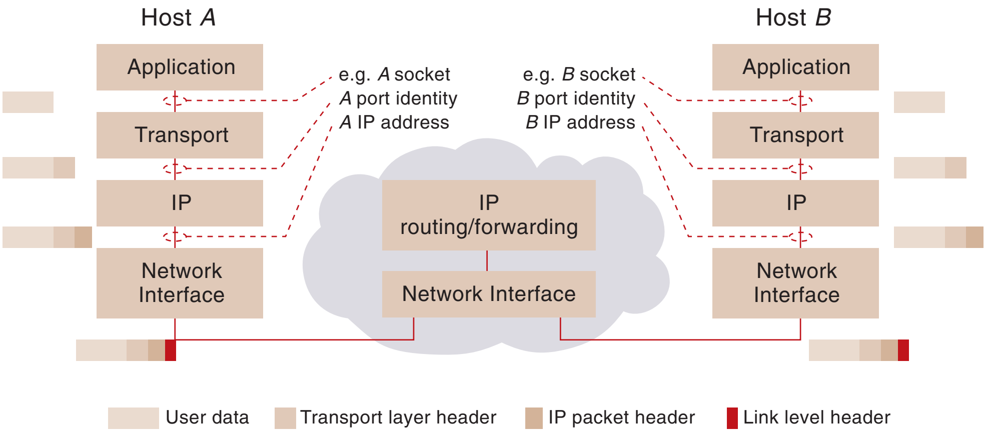


Figure 7 Hierarchy of functionality/protocols

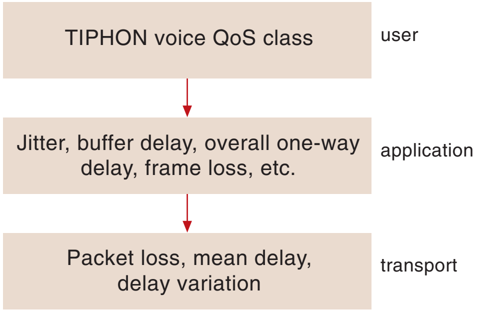

TIPHON voice QoS class
user
Jitter, buffer delay, overall one-way delay, frame loss, etc.
application
Packet loss, mean delay, delay variation
transport

Figure 8 Layers for TIPHON, a telephony service, from [TS329-3]

A principle behind the layering is that a layer entity at a destination sees the same object/message as sent by the corresponding layer entity in the source as illustrated in Figure 7.

An application interacts with a transport protocol. The application chooses the format of data transfer. Two examples are stream (a longer flow of information) and transaction (an exchange of a single information unit). In the transport layer an association with the transport layer at the destination is established (frequently called end-to-end, although when interworking units are involved the user information could even be carried further before the final destination is reached). The transport protocol divides the data into units, sometimes called segments, and transfers these units to the IP layer. In the IP layer, these data units are enveloped by the IP header information and transferred to the network interface. In a terminal/host that interface would be interacting with hardware drivers.

Even QoS parameters can be assigned to layers as illustrated in Figure 8. This implies that different aspects can be discussed on certain layers, for example allowing for “hiding” characteristics of lower layers. However, it also raises the need for finding the mapping between parameter values on the different layers. When fairly generic layers are used, such a mapping may not be straightforward, balancing the requirements of the upper layers with efficient utilisation of resources seen by layers below.

3 Optics and IP

Several questioning to what extent the traffic load growth in the core of IP-based networks is hindered by the access link capacity (e.g. dial-up, ISDN, GSM). Introducing access links with higher capacity, the traffic loads in the core networks may grow even more drastically. This is one argument for investigating use of optics in closer connection with IP (although the general traffic growth and price trends for optics also advocate this).

One of the means to step up the traffic handling capability of the IP network is to develop routers with higher throughput. Some means undertaken are:

- Separate forwarding and route determination and make the routing software leaner;
- Introduce interfaces with higher transfer rates, increased switching speed;
- Introduce hardware adapted solutions (e.g. through application-specific integrated circuits, ASICs).

Making leaner software solutions may in some respects be contrasting the functionality for traffic handling according to the TE mechanisms. Reducing the number of layers is one step to reduce the overhead. Hence, IP “directly” over optics has become a theme gaining more interest.

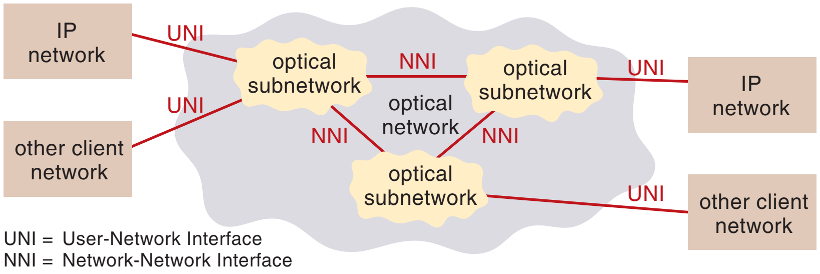


Figure 9 Schematic illustration of optical network with client networks

3.1 Interconnecting IP and Optics

The optical networks must be survivable, flexible and controllable [ID_IPoptfw]. Introducing more "intelligence" in the control plane for the optical networks is a step in doing this. As IP is seen as being a common protocol for much of the traffic carried by the optical network, utilising similar mechanisms as found for the IP-based networks is looked at with increasing interest.

A first issue is the adaptation and reuse of IP control plane protocols for the control plane in optical networks. These are to be used no matter which traffic flows (IP or non-IP) that are carried. A second issue is how IP traffic can be carried where joint control and co-ordination between the IP and optical layer are utilised.

A schematic illustration is depicted in Figure 9. An optical subnetwork may consist of all-Optical Cross-Connects (OXCs) or some nodes where optical-electrical-optical conversion is used. Two types of control interfaces are indicated; User-Network Interface (UNI) between the clients and the optical network, and Network-Network Interface (NNI) between optical subnetworks. The control flow across the UNI would naturally depend on the services offered to the client. As the NNI control flow would be derived from IP control, similarities between NNI and UNI may well exist when an IP network is the client. The physical implementation of the UNI may vary, such as:

- Direct interface with an in-band or out-of-band IP control channel. This channel is used for exchanging signalling and routing messages between the router and the OXC (like a peering arrangement);

- Indirect interface with out-of-band IP control channel. The channel may be running between management systems or servers;

- Provisioned interface involving manual operations.

Two service models are outlined in [ID_IPoptfw]:

- Domain service model where the optical network primarily offers high bandwidth connectivity (services like light-path creation, deletion, modification and status enquiry);

- Unified service model where the IP and the optical network are treated together as an integrated network. Then the OXCs will be treated like any router as seen from the control plane. No distinction is then made between UNI, NNI and any other router-to-router interface. Such an interface is assumed to be MPLS-based.

It is important to make a separation between the control plane and the data plane over the UNI. As mentioned, the optical network basically provides services to clients in the form of transport capacities (by light-paths). IP routers at the edge of the optical network must establish such paths before the communication at the IP layer can start. Therefore, the IP data plan over optical network is done over an underlying network of optical paths. On the other hand, for the control plane the IP routers and the OXCs can have peering relations, in particular for routing information exchanges. Various degrees of loose or tight coupling between the IP and the optical network may be used. The coupling is given by the details of topology and routing information exchanged, level of control that IP routers can exercise on selecting specific paths, and policies regarding dynamic provisioning of optical paths between routers (including access control, accounting and security).

Three interconnection models are then seen:

- *Overlay* model: The routing, topology distribution, signalling protocols are independent for the IP/MPLS and the optical network.

- *Augmented* model: Routing instances in the IP layer and the optical network are separated by information exchanged (e.g. IP addresses are known to the optical routing protocols).

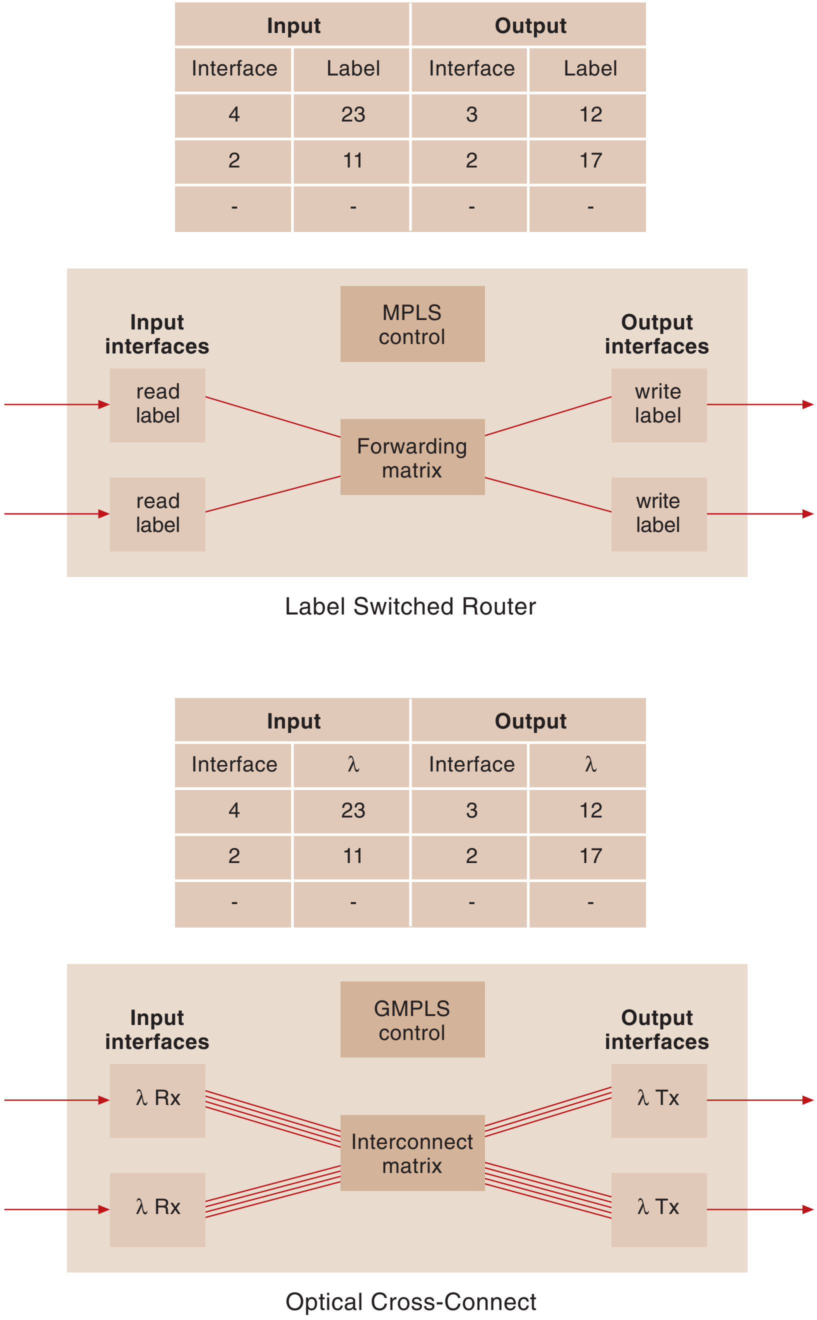
- *Peer* model: The IP/MPLS layers act as peers to the optical network. Then a single routing protocol instance can be used for the IP/MPLS network and the optical network.

These models refer to a certain degree of implementation complexity; the overlay being the least complex one for near-term deployment and the peer model the most complex one. As each of the models has its advantages, an evolution path for IP over optical network may be seen.

The migration path described in [ID_IPoptfw] is to start with the simpler functionality, meaning the domain service model with overlay interconnection and no routing exchange between the IP and the optical network. A provisioned interface would be expected. The next phase of the migration path is to exchange reachability information between IP and the optical network. This may allow for light-paths to be established in conjunction with setting up LSPs. The third phase of the migration is to support the peer model.

Applying a common signalling framework from the start would assist the migration. For the domain service model, implementation agreement based on Generalised MPLS (GMPLS) UNI signalling is being developed by the Optical Interworking Forum. This is intended for near-term deployment, although helping in the migration toward the peer model. This is said to support incremental development as the interconnection model increases in complexity.

The GMPLS is described in [ID_GMPLS]. This basically contains extensions to signalling for MPLS, the need to include time-division, wavelength and spatial switched/divided systems, see illustration in Figure 10.



Input		Output	
Interface	Label	Interface	Label
4	23	3	12
2	11	2	17
-	-	-	-

Input		Output	
Interface	λ	Interface	λ
4	23	3	12
2	11	2	17
-	-	-	-

Figure 10 Similarities between MPLS and GMPLS for a Label Switched Router and an Optical Cross-Connect

3.2 Multiplexing Hierarchy

As described in [Jens01], MPLS uses labels to support forwarding of packets. Label Switching Routers (LSRs) have a forwarding table recognising the cells/frames with the labels, or the IP packet headers (at the border of the MPLS domain). This is extended in GMPLS where the following interfaces are given for an LSR:

- Interfaces that recognise packet/cell/frame boundaries and forward the data based on the content in the packet or label/cell header. This is referred to as *Packet-Switch Capable*. Examples are MPLS-capable routers and ATM switches.

- Interfaces that forward data based on time slots in a periodic cycle. This is referred to as *Time-Division Multiplex Capable*. An example is an SDH cross-connect.

- Interfaces that forward data based on the wavelength. Such interfaces are referred to as *Lambda Switch Capable*. An optical cross-connect is an example.

- Interfaces that forward data based on physical space position of data. This is referred to as *Fibre-Switch Capable*. An optical cross-connect operating on the level of single or multiple fibres is an example.

These can be organised in a hierarchical manner as shown in Figure 11 and corresponding labels and Label Switched Paths (LSPs) defined. Then an LSP that starts and ends on a packet-switch capable interface can be grouped together with other similar LSPs into a common LSP that starts and ends on a time-division multiplex interface. This LSP can be grouped together with other similar LSPs into an LSP that starts and ends on a lambda switch capable interface, and so forth. This is similar to a multiplexing hierarchy.

Compared to MPLS, the GMPLS introduces additional interface types. The formats of the labels on the interfaces are given in [ID_GMPLS].

Motivations for combining solutions for MPLS, in particular related to Traffic Engineering, and mechanisms for control plane in OXCs are described in [ID_MPLSoptte]:

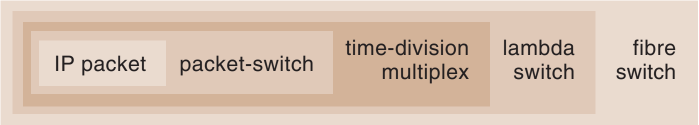


Figure 11 Organising "labels" hierarchically

- To provide a framework for real-time provisioning of optical channels in automatically switched optical networks;
- To foster the expedited development and deployment of a new class of versatile OXCs;
- To allow the use of uniform semantics for network management and operation control and hybrid networks consisting of OXCs and label switching routers (LSRs).

A particular emphasis may be placed on the support of various protection and restoration schemes.

3.3 Traffic Engineering Topics

The main components of the MPLS TE control plane model include (ref. [Jens01]):

- Discovery of resources;
- Dissemination of state information (similar to abilities of routing protocols);
- Selection of paths (similar to constraint-based routing);
- Management of paths (label distribution, path placement, path maintenance, path revocation).

Several of the capabilities can be derived from MPLS by replacing "traffic trunk" with "optical channel".

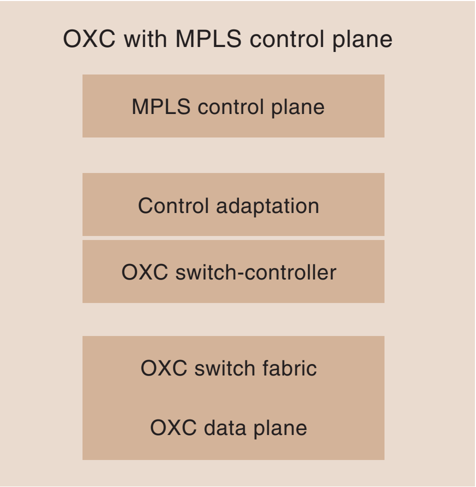


Figure 12 Potential OXC system architecture, adapted from [ID_MPLSoptte]

3.4 Optical Transport Networks

A high level architectural model is presented in [ID_MPLSoptte]. The modelling aspects have been grouped into a horizontal dimension and a vertical dimension. The horizontal dimension refers to special requirements for an Optical Transport Network (OTN) including considerations such as:

- Type of OTN state information should be discovered and disseminated to support path selection for optical channels (e.g. attenuation, dispersion);
- Infrastructure used for propagating the control information;
- Computing constrained paths fulfilling performance and policy requirements;
- Domain specific requirements for establishing optical channels and enhancements for MPLS signalling protocols for addressing these requirements.

The vertical dimension includes concerns when porting MPLS control plane software onto an OXC. A potential architecture of an OXC is also given in [ID_MPLSoptte], see Figure 12.

Looking closer into an OTN as described by ITU-T, it should be noted that it is itself divided into layers, including:

- an optical channel (OCh) layer network;
- an optical multiplex section (OMS) layer network;
- an optical transmission section (OTS) layer network.

4 Mobility

4.1 Mobile Routing – Mobile IP

As users and hosts may be moving around and still want to be connected to a network as if they were "at home", the network has to be capable of supporting this. One issue is mobile routing where a mobile user (mobile node) may move while still having a predefined home location (where a *Home Agent*, HA, is located).

Mobility support is described in [RFC2002]. The assumptions made are:

- No additional constraints on the assignment of IP addresses;
- The mobile user will not change points of attachment very frequently, say less frequent than once per second;

- IP packets are routed based on the destination IP address.

By allowing the mobile node to use two IP addresses the mobility is supported by IP. Hence, one of the addresses is fixed (indicating the node's home region), while the other address may change at each point of attachment (called a *care-of-address*).

Two types of care-of-address can be used: a "foreign agent care-of-address" is the address of the *Foreign Agent* (FA) where the mobile node is registered, and a "co-located care-of-address" being an externally obtained local address that the mobile node has associated with its network interface. Hence, the care-of-address refers to the end point of the tunnel from the home agent.

Mobile IP consists of three basic mechanisms:

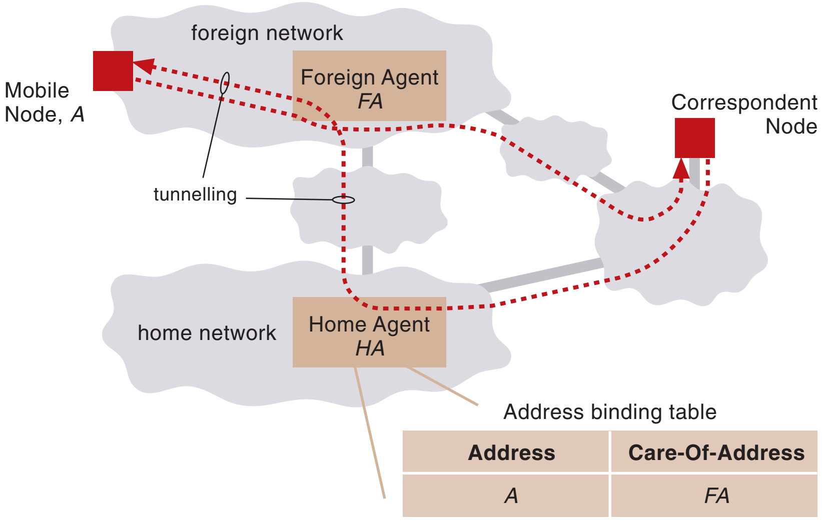
- Discovering agents and obtaining a care-of-address;
- Registering the care-of-address;
- Tunnelling to the care-of-address.

An illustration of mobile IP is given in Figure 13 where the mobile node *A* communicates with the node denoted as correspondent node.

A mobile node would belong to a home "domain", controlled by a home agent. The HA can control the forwarding of packets when the mobile node is not connected to its home domain. The mobile node also needs a Care-Of-Address (COA) in the foreign domain it is connected to. This is assigned by a foreign agent. Foreign agents are defined for each area. When the user turns up in an area, it makes its presence known (by listening for foreign agents or checking for one). The discovery messages applied are quite similar to ICMP router discovery messages.

Then the foreign agent checks with the home agent concerning authentication, etc. at the same time as telling of the mobile user's whereabouts. A care-of-address can be allocated at the same time, to be used for forwarding packets to the user. When a packet destined for the mobile user arrives at its home location area, the home agent may forward it to the foreign agent (within a tunnel), which sends it to the mobile user. At the same time the home agent may inform the sender of the packet of the new location of the mobile user such that the sender can transmit any subsequent packets directly to that area (possibly in a tunnel to the foreign agent, by using the care-of-address).

Most of the messages, as described in [RFC2002], are carried by UDP.



Address	Care-Of-Address
A	*FA*

Figure 13 Mobile IP illustration

There are multiple options for realising this scenario, like all subsequent packets may also pass through the home agent and the mobile user may be assigned a temporary address allowing it to receive all packets directly without further assistance of the foreign agent. This is then using the second type of care-of-address (co-located).

In order to avoid addresses being kept even after the mobile user has left a foreign area, the registration is only valid for a given time interval, requiring periodical refreshing. In case a number of mobile users are following a common means of transportation (like an aeroplane or train) containing a router itself, several levels of tunnelling can be used where one level refers to the on-board router.

The binding between the home address (e.g. *A*) of the mobile node and its COA (e.g. *FA*) is kept by the home agent. The binding will likely be attached with a validity duration, implying that binding updates should be initiated. A central function is for the mobile node to detect that it has moved to another domain, like using router discovery and neighbour unreachability detection.

In this way the mobile node will be accessible via its home agent. When the mobile node is connected to its home domain, normal routing is used. When a packet arrives at HA for which a valid binding is given, the HA tunnels the packet to the COA. When a mobile node receives a new COA, the information on this address is sent to its home agent.

Routing from the mobile node to the correspondent node will follow normal routes. In the opposite direction, packets may initially pass the home agent. When the mobile node receives the packets, it will learn that the correspondent node does not know its COA. In such cases, it may distribute its address to the correspondent node for more efficient routing.

For IPv6 the routing header option can be utilised in combination with mobile IP to implement more efficient routing. When the mobile node receives a new COA, the address can be distributed to its HA using the destination header option. Some essential changes in IPv6 are (see [Pain01]):

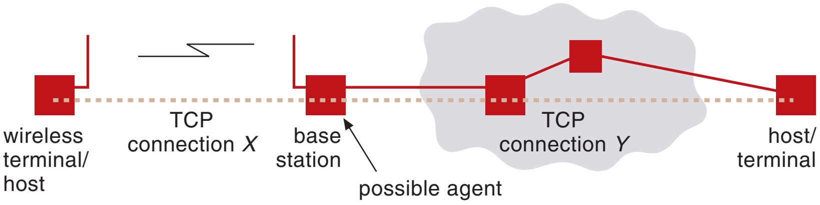
- Route optimisation: the correspondent node can send packets to the mobile node without passing through the home agent (avoiding triangular routing).
- Filtering: allowing the mobile node to use its care-of-address in the source address, any filtering in intermediate routers may be passed more easily (compared to when a "foreign" address would be used). The home address would be carried in a destination header option.
- Foreign Agents not needed: by using IPv6 features like neighbour discovery and address autoconfiguration, the FAs are eliminated.
- Security: IPv6 would use IPSec for security requirements.
- IPv6 routing headers: introducing the routing header option, IP tunnelling could be obsolete, reducing the overhead.

Mobile IP originally primarily addressed "slowly" moving entities, which is likely to be combined with other features for so-called micro-mobility. Some of these are surveyed in [Pain01].

4.2 TCP and Wireless

A transport protocol like TCP should in principle be independent of the underlying layers. However, considering the congestion control algorithm used in TCP it turns out to be sensitive to the characteristics of those layers. For instance, when a segment is lost TCP assumes this is due to network congestion and reduces its rate of transmission. When a system introducing high bit error rate is used, a segment could very well be "lost" as seen from the TCP layer although all bits have been received (albeit some bits incorrectly). In such a case, a proper behaviour of the TCP layer may not be to reduce its rate, but rather to keep, or even increase its rate. For wireless systems, one suggested solution is to divide the TCP connection into two parts, each of the parts having separate characteristics, see Figure 14. One of the parts traverses the wired part where lost TCP segments commonly imply congestion asking for the regular behaviour of TCP.

Figure 14 Dividing into several TCP connections



The other part may ask for a different behaviour, implying that the TCP functions should be changed. In this case the base station has to terminate both TCP connections, meaning that end-to-end connection and acknowledgements are not present.

Another suggestion is not to split the TCP connection, but to introduce an agent into the base station. This agent will examine the TCP segments transmitted on the air interface and retransmit the segment if an acknowledgement has not been received within a short delay. This is to "hide" those losses to the original sender, which after a relatively longer delay will retransmit the packets. In case there is a high bit error ratio on the air interface, the retransmission timer in the original sender may expire, after all resulting in little gain. This solution could be further extended by introducing selective retransmissions, suppression of acknowledgement duplicates and so forth.

Another issue is to deal with handovers. Two factors are that the handovers may take some time, and that the effective throughput may be quite different on the interfaces before and after the handover. An example of the latter is handover from a LAN to a GSM network, e.g. for an ongoing file transfer. Hence, combining wireless and transport protocols may imply additional challenges.

5 Multicast

Multicast, as the term says, is traffic flowing from one to many users at the same time. For such applications the use of unicast, implying sending the same information in parallel to the receivers, would in most cases be rather inefficient.

Two essential issues for multicast are:

- How is the group of receivers identified and maintained;
- How is the tree for distributing information built.

Two variations of distribution trees are source trees and shared trees, see Figure 15.

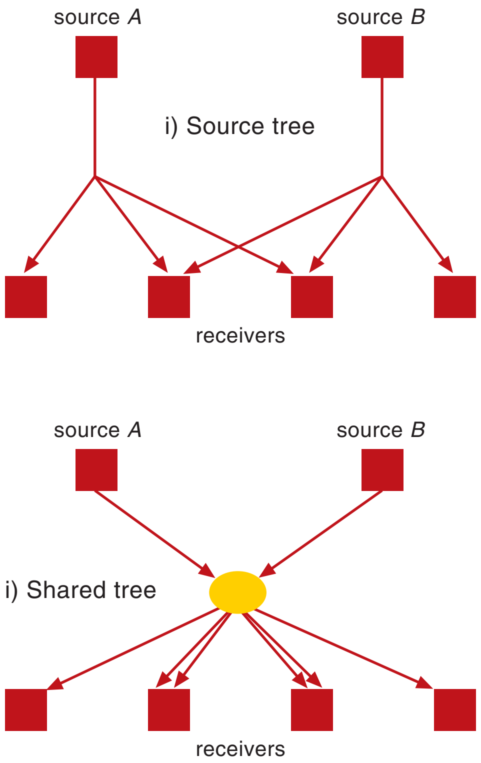


Figure 15 Source tree and shared tree

A source tree means that each source has its tree. When a number of sources share the distribution tree, this is naturally referred to as a shared tree. In the shared distribution tree, the traffic flows from all sources must go to the shared tree root, from which the information is distributed. Multicast sessions are identified by a special multicast address and all packets from the source to the receivers carry this address.

In the following some multicast protocols are described.

The Distance Vector Multicast Routing Protocol (DVMRP) and Protocol-Independent Multicast – Dense Mode (PIM-DM) use a source distribution tree and basically work on the assumption that every sub-net in the network should receive the multicast traffic. Routers that do not have any receivers interested in a multicast session respond to a DVMRP or PIM-DM request by a prune message. This implies that these sub-nets are removed from the multicast distribution tree. Hosts that want to joint a multicast session or to leave it use Internet Group Management Protocol (IGMP) messages.

The Protocol-Independent Multicast – Sparse Mode (PIM-SM) uses a shared tree that assumes that multicast traffic is not to be distributed unless specifically requested by a node. A host sends a join message when it wants to take part in a multicast session.

Compared to ordinary multicast, when the number of receivers is fairly low, is may be more efficient to explicitly list the receivers in the packet. This is proposed for Small-Group Multicast (SGM). Then every router that gets the packet has to look at the header to decide whether it should forward the packet on more outgoing interfaces.

An overview of some multicast routing protocols is given in Table 1 (from [ID_MPmc]).

Aggregation refers to whether different destination addresses are aggregated into one entry in the routing table. *Flood and prune* refers to cases when multicast protocols flood the network with multicast data. Then, some branches not used can be pruned if the nodes do not longer want to receive data any more (pruning starts from the point of the branch were no receivers are active). This allows for dynamic distribution trees. *Tree types*, being source or

	DVMRP	MOSPF	CBT	PIM-DM	PIM-SM	SSM	SM
Aggregation	no	no	no	no	no	no	no
Flood and prune	yes	no	no	yes	no	no	option
Tree type	source	source	shared	source	both	source	shared
State co-existence	no	no	no	no	yes	no	no
Uni-/bi-directional	NA	NA	bi	NA	uni	uni	bi
Encapsulation	no	no	yes	no	yes	no	yes
Loop free	no	no	no	no	no	no	no

Table 1 Characteristics of some routing protocols (DVMRP = Distance Vector Multicast Routing Protocol, MOSPF = Multicast extensions to OSPF, CBT = Core Based Trees, PIM-DM = Protocol Independent Multicast – Dense Mode, PIM-SM = Protocol Independent Multicast – Sparse Mode, SSM = Source Specific Multicast, SM = Simple Multicast)

shared, are explained above. *Co-existence* indicates whether both tree types can exist and a switchover might be possible for the same multicast session. *Uni-/bi-directional* refers to whether a shared tree supports uni- and/or bi-directional connections. *Encapsulation* indicates whether data between the source and the root node (for shared trees) is encapsulated (i.e. IP-in-IP). *Loop free* refers to whether or not loop detection is part of the multicast protocol.

6 Authentication, Authorisation, Accounting and Security

Increasing commercialisation leads to a steadily growing emphasis on the issues addressed in this chapter. Authentication, authorisation and accounting (AAA) are essential functions of network management and when interfacing customers and other operators/providers.

As a customer is eventually to pay for a service, being sure that the service is delivered to the proper party and charged for correctly is essential. Besides, having traffic flows from different parties in the network also requires adequate security mechanisms.

Authentication is not specifically described in the following. Authentication is commonly understood as confirming that the source/entity is the one it claims to be. This is often implemented by using passwords, certificates and so forth.

6.1 Authorisation Framework

Authorisation is the function of deciding whether a particular right can be granted to the presenter of a particular credential; for instance, if a given user is allowed to use a certain resource.

The framework identifies the conceptual entities that may be participants in an authorisation procedure (see Figure 16):

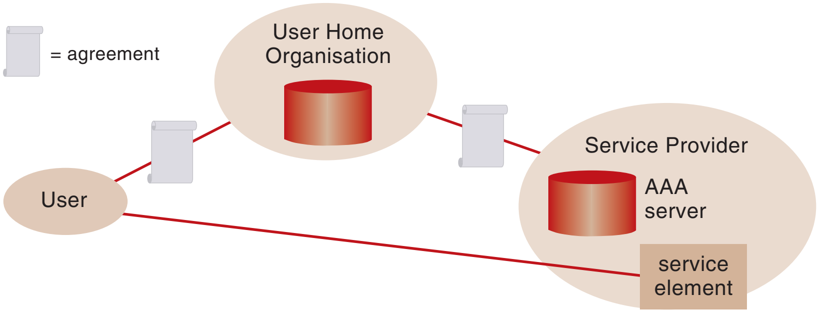
1. A User who wants to access the service or resource;

2. A User Home Organisation (UHO) that has an agreement with the user and checks whether the user is allowed to obtain the requested service or resource;

3. A Service Provider's AAA Server that authorises a service based on the agreement with the UHO without specific knowledge of the individual user;

4. A Service Provider's Service Element that provides the service itself.

Figure 16 Entities in the authorisation framework



Several scenarios are possible:

- Single domain case: the UHO and the Service Provider are the same entity. An example of this is a router controlled by a local bandwidth broker acting as the AAA server.

- Roaming: the UHO and the Service Provider are different. Their AAA servers have to co-operate in order to complete the authorisation process. An example of roaming is a Mobile IP provider allowing access to a user from another domain.

- Distributed Service: to complete a service, offerings from several service providers may need to be combined. Again, the AAA servers of the service providers have to co-operate.

In all scenarios SLAs would exist between the actors, which have to be taken into account when making authorisation decisions.

All these entities may interact in many different ways depending on the type of service and scenario. In some cases the user may send the service requests to the AAA server, while in others the request is sent to the service element (e.g. dial-in access). Also, it is possible for the user to get a ticket or certificate from the AAA server to include it in the request to the service element.

One view of an authorisation is that it is the result of evaluating policies of each organisation that has an interest in the authorisation decision. The authorisation process can be modelled in terms of the Policy Framework [Jens01a]. AAA servers may act as Policy Retrieval Points (PRP) and Policy Decision Points (PDP). Service elements correspond to Policy Enforcement Points (PEP). Both entities are also Policy Information Points (PIP) containing information needed for policy evaluation, which can be accessed as Policy Information Base (PIB). The user may also be a PRP, a PIP and a PDP if policy is used to request the service. These are described in [Jens01a].

In many applications, authorisation results in establishing an ongoing service which is called a session. Each of the AAA servers involved in the authorisation may have a Resource Manager component to keep track of the state of the session and be able to affect changes to the session if required. Resource Managers may keep complex cross-administrative domain information supported by dialogues with peer Resource Managers.

6.2 Accounting Management

Accounting is the collection of resource consumption data. Hence, accounting management requires that resource consumption is measured, rated, assigned and transferred between appropriate parties. Accounting data is needed for purposes such as trend analysis and capacity planning, billing, auditing and cost allocation.

The accounting management architecture involves interactions between routers, accounting servers and billing servers. Network devices collect resource consumption data in the form of accounting metrics. This information is then transferred to an accounting server by means of a protocol such as SNMP, see Figure 17. The accounting server may process the received accounting data to produce session records. The processed data is then transferred to a billing server, which handles rating and invoice generation, but may also carry out auditing, cost allocation, trend analysis and capacity planning. Note that some sources operate with mediation and charging levels as well.

6.3 Security

Several security mechanisms can be implemented. In this section IP secure (IPsec) and firewalls are outlined. While IPsec operates on the IP level, another applied mechanism, the Transport Layer Security (TLS) – formerly Secure Socket Layer (SSL), is applied on the TCP layer.

6.3.1 IP secure

Currently, most security concerns are taken care of by the applications. IP secure (IPsec) is a family of protocols, procedures and cryptographic algorithms that provide security services for traffic at the IP layer in both the IPv4 and IPv6 environments. The services provided are: access control, integrity, data origin authentication, protection against replays, confidentiality, and limited traffic flow confidentiality.

IPsec is based on two security protocols: Authentication Header (AH), which provides integrity, data origin authentication and anti-replay service, and Encapsulating Security Payload (ESP), which may provide either confidentiality or integrity, authentication and anti-replay.

AH is a new header subfield, which can be inserted into IPv4 or IPv6 packets. The authentication is calculated over the application data and the IP header fields (fields not being changed during the forwarding (e.g. omitting the TTL field).

EPS is a new header to be inserted in front of the original IP packet header. Hence, the total original IP packet can be encrypted. Both AH and ESP can be applied on the same packet.

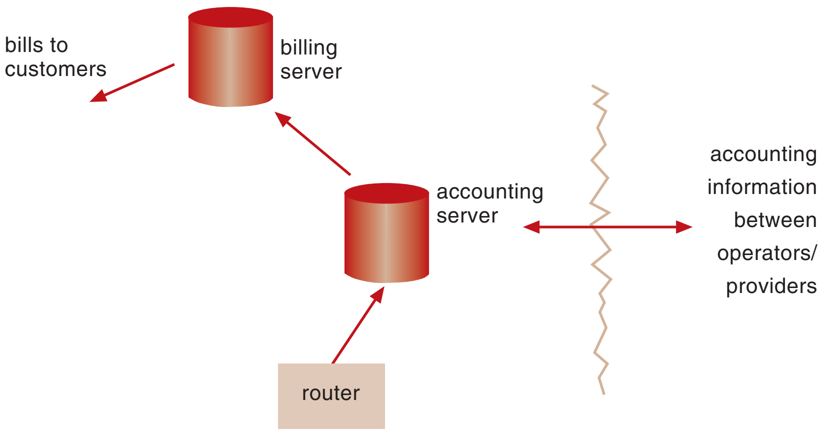
The IPsec security model is based on Security Associations (SA). An SA is a simplex, i.e. unidirectional connection that allows security services to the traffic carried by it. If traffic should be protected by both protocols, it must be processed by two SAs in sequence.

A unidirectional security association is established between a sender and a receiver. The association is identified by a Security Parameter Index (SPI) and the receiver address. The SPI is defined by several parameters including the authentication and encryption algorithms, keys and association life times. Each association is unidirectional which means that a bidirectional connection needs one security association in each direction.

As mentioned above, the authentication header offers both data integrity and authentication of IP packets. In IPv6, the authentication header includes a length field, an SPI and the authentication data. The authentication algorithm is calculated over the entire packet, excluding protocol fields that are modified in intermediate routers. Authentication is done between the sender and the receiver, or between the sender and a firewall.

The ESP provides data integrity and privacy to the users. The ESP header starts with a length field and a 32 bit SPI. The rest of the header, if

Figure 17 Example of servers and information involved in accounting and billing



present, contains parameters that depend on the encryption algorithm used. The DES – CBC[1] algorithm is mentioned as one possibility. Parts of the ESP header including the SPI are transmitted unencrypted. This privacy mechanism can be used in two ways depending on the required level of privacy. The first option is to only encrypt the transport layer segment in the IP payload. This scheme is called transport mode ESP. The other option is to encrypt the entire IP packet and encapsulate it in a new IP packet. This is called tunnel mode ESP. Transport mode offers confidentiality to the higher layer protocols by introducing little overhead. A disadvantage is that traffic analysis can be carried out by a (unwanted) third party as the packet is addressed to its final destination.

Tunnel mode ESP has more overhead than the transport mode but it prevents traffic analysis. Different key management solutions are possible, including both manual and automatic ones.

IPsec is not related in principle to QoS protocols, procedures or management. However, some aspects of IPsec may have an impact on QoS:

- The AH and ESP protocols introduce overhead for IP packets.

- Key management can increase the time to establish a connection and introduces some additional traffic.

- Cryptographic algorithms can be rather CPU consuming.

- Encryption may prevent effective compression by lower layers. To minimise this problem IPsec supports negotiation of IP compression.

- The ToS and Class fields of tunnelled packets are copied to the outer IP header, making IPsec transparent to QoS mechanisms based on the analysis of such fields.

- Any other QoS mechanism based on the inspection of fields of upper layer protocols may become useless when encryption is used.

6.3.2 Firewalls and Proxies

IPsec does not protect against every type of attack a system may be exposed to. A critical question is how internal (protected) traffic and resources can be left unexposed to external parties, and thereby avoiding that network information can be used in further attacks. Such problems are hindered by controlling all traffic entering and leaving the system. For this purpose, firewalls are introduced.

A firewall is an implementation of an access control policy between two networks. Two types of firewalls exist:

- Network level: packets are filtered on the basis of source address, destination address and port. This means that a router may be used as a network level firewall.

- Application level: a proxy server which is a software running on the firewall allowing no direct traffic between the networks. It is not transparent for applications which have to be configured to use the proxy to reach the network. One step is to perform network address translation, which hides internal addresses from the outside.

Security restrictions imposed by firewalls may make it difficult to establish end-to-end connections. In the case of network level firewalls, it is a matter of firewall configuration to allow or block the exchange of packets. Inspection of QoS-related IP header fields such as ToS or Class should then be supported.

Since proxies block all direct traffic between networks, special mechanisms must be implemented on proxy servers to support QoS guarantees. Applications should be able to inform the proxy of the required QoS parameters for the session and then the proxy should be able to establish the requested session with the remote host.

7 Scenarios/Examples

7.1 Client – Server

A large group of applications related to the current use of IP-based networks can be categorised according to the *Client – Server model*. Here the term server refers to any program that offers a service. Servers accept requests, perform their service, and return the result to the requester. A program becomes a client when it sends a request to a server and waits for a response. Commonly, the server has a well-known port that requests are using, see Figure 18. The client can allocate an unused port to this communication.

Telnet allows a user to establish a TCP connection to another machine. Then the keystrokes are passed to the remote machine and the response is

[1] *Data Encryption Standard – Cipher Block Chaining.*

commonly returned from the remote machine to the local machine, see Figure 19. The Telnet recommended setting is ToS = 1000, i.e. minimising delay [RFC1700].

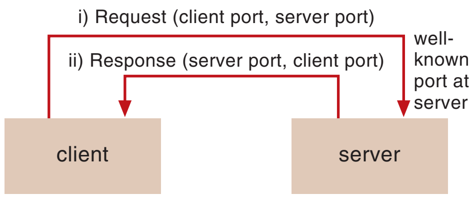

i) Request (client port, server port)
ii) Response (server port, client port)
well-known port at server
client
server

Figure 18 Client-server message exchange principles

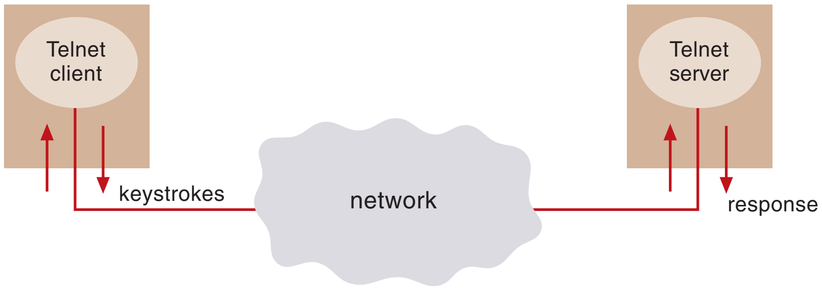


Figure 19 Telnet session between client and server

Another traditional application is the file transfer protocol (*FTP*). By this a user can log onto a remote machine and handle files (in addition to a few limited commands). FTP may establish several TCP connections, e.g. one for control and another for data transfer (Telnet can then be used for the control session).

The FTP recommended setting is ToS equal to 1000 for control and 0100 for data flow (maximise throughput for the data flow).

7.2 Virtual Private Networks – Provider-based

7.2.1 Overview

A Virtual Private Network (VPN) refers to an interconnection of customer sites, making them appear like a common network, where the interconnection is done by using resources in a shared (public) network. A framework for VPN is described in [RFC2764]. There the term VPN refers to the emulation of a private Wide Area Network (WAN) facility using IP facilities (including the public Internet or private IP backbone). Hence, the VPN is considered as a connectivity object, where hosts/terminals are attached.

The logical structure of the VPN, like addressing, reachability and access control, is the same as if the sites were connected by private lines.

A provider-provisioned VPN refers to a VPN where the service provider participates in management and provisioning of the VPN.

An illustration is given in Figure 20, containing Customer Edge (CE) devices, Provider Edge (PE) routers and Provider (P) routers. In several cases, customers may use private addressing space, implying that IP addresses would not be globally unique. This means that a PE router that connects several different customer networks might have different addressing schemes for each network (unless the tunnelling is started in the CE devices). The use of tunnelling is further advocated by a level of isolation between the packets from different customer networks having to be maintained.

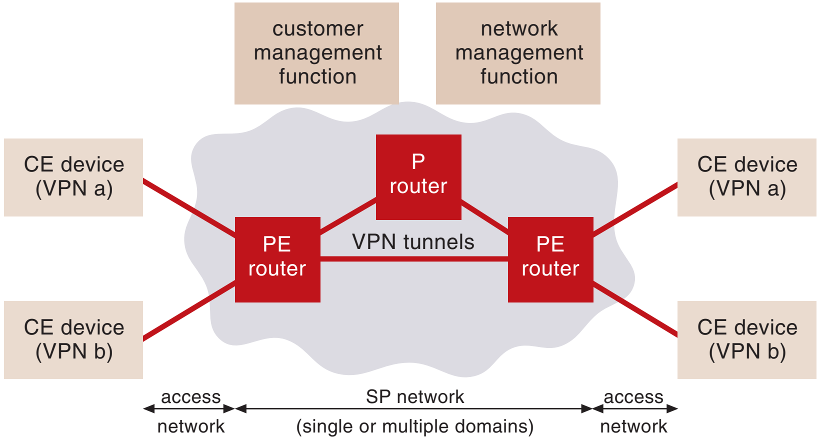
Two main types of VPNs are described in [ID_ppvpnfw]:

- CPE-based VPN (Customer Premises Equipment): Knowledge of the customer network is only given in the customer equipment, hence the service provider is not aware of it. Then, the customer network is supported by tunnels set up between CPEs.

- Network-based VPN: Routers in the SP network provides the VPN, which may allow for hiding the VPN from the customer equipment. Then, the customer networks are supported by tunnels set up between PE routers.

The network-based VPNs are commonly referred to as provider-provisioned VPNs. Depending on the interconnection offered to the customer sites, at least three types can be identified: BGP-VPNs, VPNs based on virtual routers, and port-based VPNs. The latter refer to layer 2 (or layer 1) interface, like Frame Relay, ATM, SDH, etc.

Figure 20 Illustration of VPN



A virtual router approach in combination with MPLS is described in [RFC2917]. Compared to BGP-VPNs (called overlay models in the RFC), no modifications are needed for the routing protocol applied. A virtual router is described as a collection of threads, either static or dynamic in a router, that provides routing and forwarding services. These services are similar as if physical routers have been applied. A virtual router is set up to give the illusion that a physical router is present. Therefore it provides an element in the (virtual) routing domain. Hence, given that the virtual router connects to a specific (logically discrete) routing domain and that a physical router can support multiple virtual routers, it follows that a physical router supports multiple (logically discreet) routing domains, [RFC2917]. It is further stated that the following aspects of a router must be emulated:

- Configuration of any combination of routing protocols;
- Monitoring of the network;
- Trouble shooting.

Independent of VPN types a set of requirements can be identified, including security, manageability, interoperability, scalability, traffic engineering and QoS/SLA/SLS support. A Service Level Specification (SLS) may be defined for each VPN, VPN site, interface, or similar. Target values and measurement procedures for a set of parameters are typically defined, including:

- Traffic values (bit rates) and QoS values for each service class and for aggregates;

- Ways of handling non-conformant traffic;

- Availability for a site, for the VPN or for the interface;

- Duration of outage times per site, route, VPN, etc.;

- Time for activating a new service;

- Response time for trouble reporting;

- Repair time.

A VPN may carry traffic flows for several types of applications. Some flows may have real-time requirements, while others are more elastic. Hence, both the IntServ model for selected individual flows and DiffServ for aggregated flows might be requested within a VPN (see [Jens01] for description of IntServ and DiffServ). A specific requirement is that the class assigned to a traffic flow at the ingress of the VPN should be kept on the egress of the VPN (called service class transparency). An example of this is to keep the packet's assignment to the DiffServ class.

Different types of encapsulation may be used for the tunnels:

- MPLS, as described in [Jens01]. Labels are attached to the IP packets which give the forwarding treatment and the Label Switched Path (LSP) to follow. Several LSPs may be multiplexed into other LSPs. This requires state information per VPN. Some differentiation may be supported. LSPs may be established and maintained by signalling or management procedures.

- IPSec, as described in Section 7.3.1. Multiplexing may be supported and the Internet Key Exchange (IKE) protocol is used for establishing and maintaining protocols.

- Generic Routing Encapsulation (GRE), being a protocol for encapsulating any payload protocol over any link (delivery) protocol (e.g. IP-in-IP). Multiplexing is not supported and there are no specific procedures for establishing and maintaining the tunnels.

- IP-in-IP, referring to encapsulating IP packets within other IP packets as described in Section 3.1.

A VPN membership refers to the association of VPNs, CEs and PEs. A certain CE belongs to one or more VPNs. The set of VPNs that a PE is involved in may change over time due to added or deleted customer networks or their changed configurations. Appropriate means for distributing VPN membership information must therefore be implemented.

In case the provider network (at least PE routers) operates on layer 3 (that is examines IP packet headers), independent forwarding tables could emerge for each VPN, sometimes referred to as a VPN forwarding instance (VFI). This also resembles the virtual router concept. A VFI is a logical entity in a PE containing router information base and forwarding information base for a VPN. The VFI terminates tunnels for interconnecting with other VFIs and terminates access connections for connected CEs.

7.2.2 VPN by MPLS and BGP

A method for providing the VPN service in an IP-based backbone using MPLS and BGP is described by [RFC2547]. MPLS is used for forwarding (tunnelling), while BGP is used for distributing routing information. In this way a VPN is established which itself may provide IP services to customers (e.g. considered as a "wholesale VPN"). The common backbone can then be used for a number of VPNs. As described above, a PE router maintains a forwarding table per VPN it takes part in. If a packet arrives containing an IP destination address not matching an entry in the forwarding table, the packet could be forwarded on the "public Internet" if external access is allowed for that VPN (implying that the "public Internet" forwarding table is examined). To keep VPNs isolated only packets/ labels belonging to a given VPN must be accepted and forwarded according to that VPN's rules.

A two-level MPLS label stack is used in the backbone, see Figure 21. When a PE receives a packet from a CE it selects the appropriate forwarding table to use (based on knowledge of the VPN in question). If the packet is to be forwarded to another router in the backbone, a label is attached according to the BGP Next Hop information (commonly to reach the PE on the egress side as part of that VPN). This label can be called the "bottom label". Then the PE looks into the "ordinary" IGP routing and finds the IGP next hop as well as the label to assign to reach that node. This can be referred to as the "top label". In case the IGP and BGP next hops are the same a single label may suffice. BGP can then be used between the PE routers taking care of routing related to each of the VPNs, while IGP is used between the backbone routers as before.

The packet is then carried through the backbone where a P router looks at the labels to find the next hop and label to be used as explained for MPLS in [Jens01]. At the egress PE the labels are removed (even the bottom label), as the CE will only see an ordinary IP packet.

The two-level labelling allows all P routers to be unaware of the VPNs, thus supporting simplicity and scalability for those routers.

BGP-MPLS VPNs can also be applied to provide the VPN service to customers having IPv6 as outlined in [ID_BVIPv6]. Then MPLS is used to forward packets and BGP is enhanced for distribution of VPN routers.

7.2.3 Dialling up to VPNs

A dialling up feature allows a user to connect to a VPN through an ad hoc tunnel, e.g. running in PSTN/ISDN. Hence the user might get the impression to be directly connected to that VPN (although the bit rate may well be a bit lower). Accessing by use of a public network, user authentication is naturally a main requirement. This is a common solution for home-office accessing a LAN at the office buildings. Then a Point-to-Point Protocol (PPP) connection is often used between the user and the Network Access Server (NAS). In the NAS the user is authenticated, e.g. using the Radius protocol. However, the authentication may also be done by the corporate network side. Two examples are depicted in Figure 22.

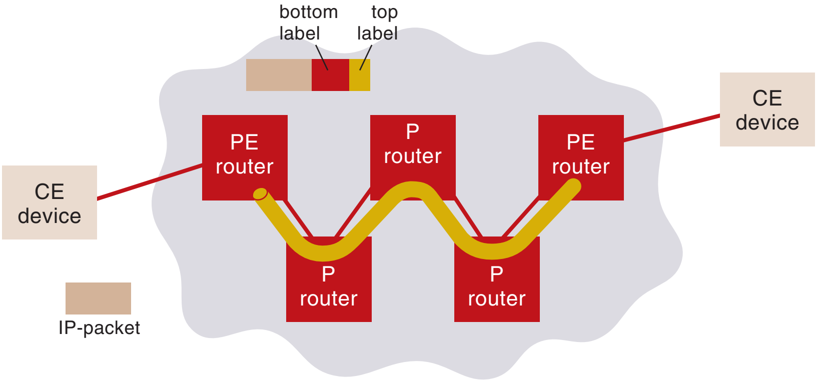
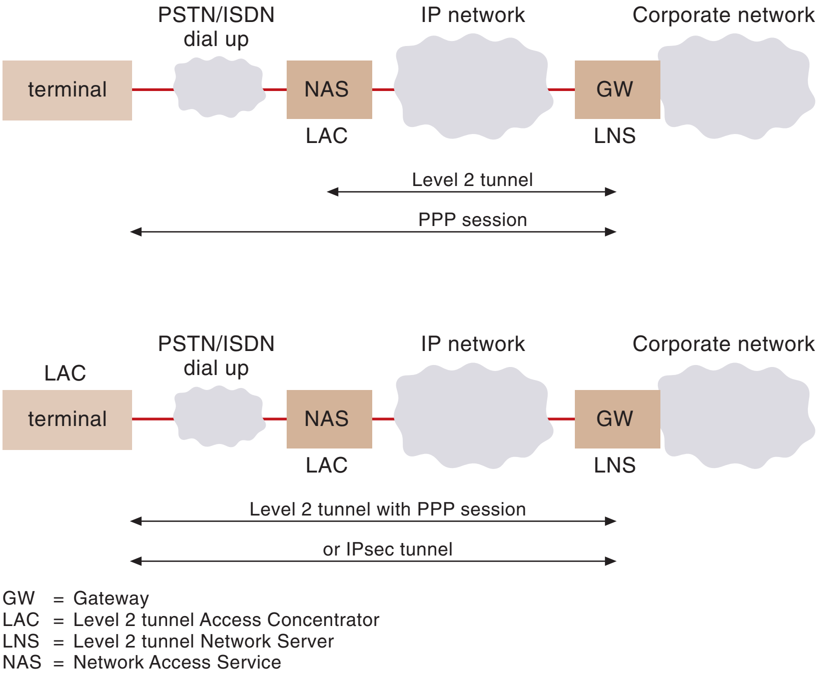


Figure 21 Illustration of two-level label

7.3 WWW

The World Wide Web (WWW) can be seen as a framework for accessing linked documents stored on various servers. Its steadily growing popularity may stem from the fact that easy to use interfaces (browser programs) are available and that a huge amount of information is stored also including colourful illustrations. WWW is basically a client – server system where the client requests information from the server. A document is commonly called a page, where each page may contain links to other pages, possibly located at other servers. By using a browser, links at the page can be clicked on which then results in downloading the requested

Figure 22 Examples of dial up configurations: compulsory configuration (upper) and optional configurations (lower)



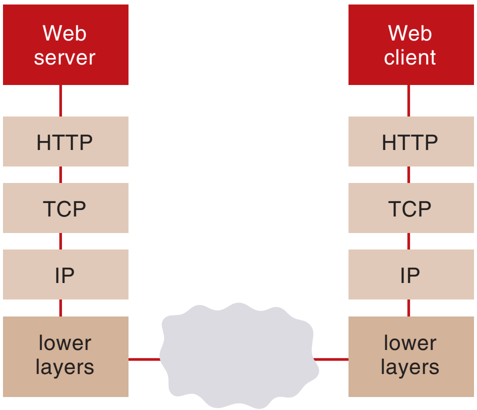


Figure 23 Some protocols used for Web surfing

page. The links at the pages are said to use hypertext.

The client is located at the user. The browser fetches the page requested, interprets the text and formatting commands that it contains, and displays the page to the user. Pages may contain text, images, sound or video tracks, etc. The browser may be assisted by a helper application to be run with the page as input. Some helper applications contain interpreters for particular languages, enabling downloading and running programs from WWW pages.

The way a browser fetches a page on a server is to establish a TCP connection to the server and then send a request on that connection.

An illustration of the protocols is given in Figure 23. Depending on the medium and traffic flows, other protocols could also be involved.

Every WWW server has a process listening to TCP port 80 for incoming connections from clients (browsers). After a TCP connection has been established the server receives the request and a response is sent. This is usually done by the HyperText Transfer Protocol (HTTP). An address of the page is commonly in a Uniform Resource Locator (URL) format. One example is *http://www.telenor.com/Telektronikk/TraffEng.html*. This consists of three parts: *http* the name of the protocol; *www.telenor.com* the server where the page is located; and *Telektronikk/TraffEng.html* being the name of the file containing the page (in case the file name is omitted, a default file is assumed, like *index.html*). This shows that the user has to know where the page is located; that is the server name. To support referencing to pages without knowing their location (as well as supporting replication) work on Universal Resource Identifiers (URIs) has been undertaken.

The DNS is used for translating the server name to the corresponding IP address. This allows the browser/client to establish a TCP connection on port 80 to the proper server. Then an HTTP message 'Get' is sent identifying the file (*Telektronikk/TraffEng.html* in the example above). The server returns this file and the TCP connection is released. The browser displays the information in the page. Several TCP connections may be established to fetch different parts of the page, like a text frame, an image frame, and so forth.

WWW pages are written in a language called HyperText Markup Language (HTML), including text, images, links to other pages, etc. By HTML can be described a "static" layout of WWW pages. Java can be applied in order to allow for more rapid interactions. Java borrows many ideas from C and C++ languages, although it is not fully compatible with any of them. The main idea is that a page can point to a Java program called an applet. When the browser receives it, the applet is placed on a client machine and executed. This allows for animation and sound as well as simplifying forms, and so forth. The applets developer writes the applet in Java and then compiles it into byte code. Then the browser needs to understand the applet and a byte code interpreter is needed in the client system. Such an interpreter may also be called a Java virtual machine. Thus, if a new data format is stored at a site, the only thing needed for someone to download and view the data is to also fetch the applet allowing the client to view the data. This may also go for protocols, which can be written in the Java language and loaded dynamically when needed.

Plug-ins are dynamically loaded code modules that become part of the browser's code. This is a flexible way to add functionality to a browser. Some plug-ins are placed in a special directory into which the browser looks. However, plug-ins can also be downloaded from a server when needed. Each plug-in commonly handles a few document types. This means that the plug-in is not using memory as long as no document of that type has been loaded.

A complete screen as observed by a user may consist of a number of pages so the browser may decide the sequence of fetching such pages. Text pages can normally be downloaded quite rapidly, also keeping the user occupied, while the more data voluminous pages are transferred, like images. Thus the user may also decide to abort the transmission if the information already available does not warrant the wait for the rest of the information. To support this, most browsers have a status line displaying which step they are currently in.

Some browsers support the use of local disk to cache pages that have been fetched. Then, before a page is fetched, the local cache can be examined to see if the page is there and if the page is still up to date. This allows a more rapid view of the corresponding pages as the information does not have to be transferred.

Considering that human end users are commonly involved and that significant amounts of data are transferred, ways to improve the service performance are strived for. One approach is to store some of the objects on the page on a different server more local to the user. This server may be dynamically selected based on current load and performance pattern. This may be called *traffic directing* as requests are directed to other servers.

Another aspect is to balance the loads among the servers, e.g. to speed up the delivery of Web-pages. How to balance the load in a distributed environment is a complex matter; one way being to introduce a load balancer which can be implemented in different terminals/servers.

A proxy sever can be seen as a gateway having multiple functions. One function can be that it accepts HTTP requests and translates these into other protocols, e.g. FTP. Another function is that the proxy server may implement a cache. A third function is access control, both for requests going out to the rest of the networks and for responses that arrive, i.e. a firewall function. Hence, a proxy is commonly present for a company network supporting Web browsing.

7.4 Supporting Telephony

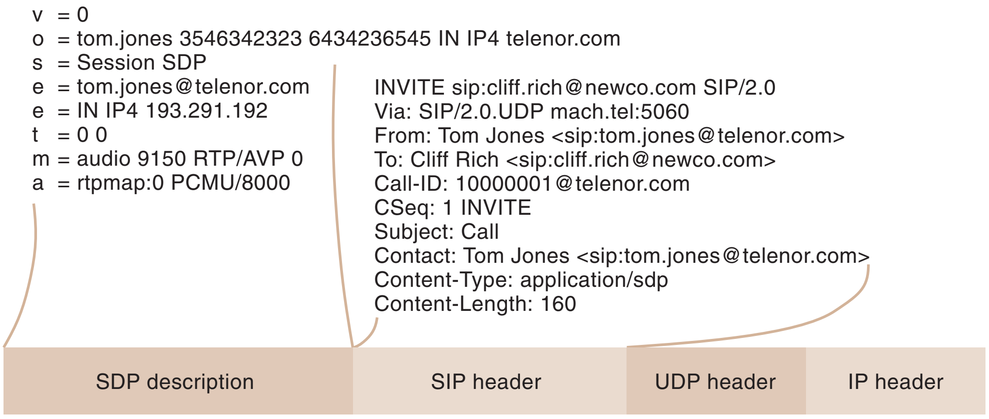
7.4.1 Protocols and Servers

The Session Initiation Protocol (SIP) is a protocol elaborated by IETF activities used for establishing, modifying and releasing real-time calls and conferences over IP-based networks. Each session may include different traffic flows, such as audio and video. SIP is a text-based and general-purpose protocol. An alternative is to use the H.323 architecture as described by ITU-T. Broadly speaking, SIP may be thought of as the call control protocol of an IP session. The basic SIP architecture may include a location data base that allows users to be contacted at the locations where they are registered. For this five aspects are considered, ref. [RFC2543]:

- User location: determination of the end system to be used for communication;
- User capabilities: determination of the media and media parameters to be used;
- User availability: determination of the willingness of the called party to engage in communications;
- Call set-up: “ringing”, establishment of call parameters at both called and calling party;
- Call handling: including transfer and release of calls.

SIP is part of the multimedia data and control architecture as depicted in Figure 31. RSVP is used for reserving network resources, the real-time transport protocol (RTP) is used for transporting real-time data and providing feedback, the real-time streaming protocol (RTSP) is used for controlling delivery of streaming media, the session announcement protocol (SAP) for advertising multimedia sessions via multicast, and the session description protocol (SDP) for describing multimedia sessions. However, it is stated that SIP does not depend on either of these.

The Real-Time Protocol (RTP) has been mentioned, which has emerged as a commonly used protocol for real-time traffic flows in IP-based networks. RTP is a protocol that provides identification of media type and synchronisation information (time stamps). These allow individual packets to be reconstructed by a receiver. Additional information is needed to support flow control and management of the traffic flows. Here the RTP Control Protocol (RTCP) has been



```
v  = 0
o  = tom.jones 3546342323 6434236545 IN IP4 telenor.com
s  = Session SDP
e  = tom.jones@telenor.com
e  = IN IP4 193.291.192
t  = 0 0
m = audio 9150 RTP/AVP 0
a  = rtpmap:0 PCMU/8000
```

```
INVITE sip:cliff.rich@newco.com SIP/2.0
Via: SIP/2.0.UDP mach.tel:5060
From: Tom Jones <sip:tom.jones@telenor.com>
To: Cliff Rich <sip:cliff.rich@newco.com>
Call-ID: 10000001@telenor.com
CSeq: 1 INVITE
Subject: Call
Contact: Tom Jones <sip:tom.jones@telenor.com>
Content-Type: application/sdp
Content-Length: 160
```

Figure 24 Example of SIP message

developed. Whenever an RTP flow is used, corresponding out-of-band RTCP flows are established, enabling the sender and receiver to exchange information on the performance of the traffic flow and may be used for higher level application control functions. Commonly, RTP is carried in UDP packets, and then the RTP session is mostly associated with an even numbered port and its corresponding RTCP associated with the next higher odd numbered port.

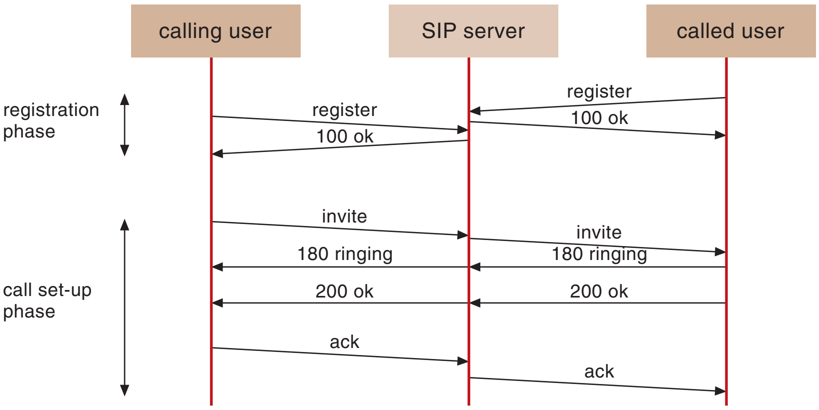
Six SIP messages have been specified: i) invite – to begin an SIP dialogue; ii) ack – to respond to an SIP request; iii) cancel – to reject the session; iv) bye – to disconnect a session after it has been established; v) options – to discover user's response without actually sending an invitation; and vi) register – to register by a location data base. In reply to a SIP message a response is generated, of which there are six types: 1xx informational, 2xx success, 3xx redirection, 4xx client error, 5xx server error, 6xx global failure. These response messages follow a format similar to the ones used for HTTP.

The format of SIP messages comprises two parts; a header consisting of SIP fields, and a body. Header fields contain parameters such as the identity of the caller, the identity of the receiver, a unique call identity, sequence number, etc.; see Figure 24. The body typically uses the Session Description Protocol (SDP) to describe the session. SIP messages are coded in plain text.

A short form (compact form) can be used to identify the fields, using a single letter coding (c – Content-Type; e – Content-Encoding; f – From; i – Call-ID; m – Contact; l – Content Length; s – Subject; t – To; v – Via).

In order to avoid that the calling user has to know the whereabouts of the called user in advance, a number of elements are included in the SIP architecture:

Figure 25 Example of registration and set-up using SIP



- Proxy servers: An intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are served internally or passed on, possibly after translation, to other servers. A proxy interprets, and, if necessary, rewrites a request message before forwarding it.

- Location servers: Location server may be co-located with another SIP server.

- Registration servers: A registrar is a server that accepts Register requests. A registrar is typically co-located with a proxy or redirect server and may offer location services.

- Redirect servers: A redirect server accepts a SIP request, maps the address into zero or more new addresses and returns these addresses to the client. Unlike a proxy server, it does not initiate its own SP request. Unlike a user agent server, it does not accept calls.

- User agent: A user agent is an application that contains both a user agent client (initiate a session) and a user agent server (receives a session request from the network)

Here, a client is an application program that sends SIP requests. The client may or may not interact directly with a human user. A server is an application program that accepts requests in order to serve requests and sends back responses to those requests.

Every SIP user has a SIP URL. They are similar to e-mail addresses, for instance sip:tom.jones@telenor.com. There is a registration server within a user's home domain. When a user is started, it sends a register message to this registration server which typically contains the URL of the user, that user's actual terminal address, port number, transport protocol (TCP, UDP), time stamp for duration of registration. The registration server authenticates the user and inserts the mapping between URL and the terminal address in the location data base. This allows users to be reached irrespective of their actual point of contact, similar to mobile IP.

So, when a calling user wants to reach the called user, assistance of a proxy or redirect server. Such servers would request mapping from URL to terminal address from the location server. Naturally, when the users already know the terminal addresses, such requests are not needed.

A simplified message sequence chart is depicted in Figure 25.

SIP is a protocol that deals with session initiation and does not explicitly describe how resources should be reserved or how security should be implemented. This is left for other protocols. In this respect, SIP is more limited in scope than for example the architecture described by ITU-T H. 323.

A client – server architecture is used. The main entities are the User Agent, the SIP Proxy Server, the SIP Redirect Server and the Registrar, some illustrated in Figure 26.

The User Agents (also referred to as SIP end-points), work as clients (UACs) when initiating requests and as servers (UASs) when responding to requests. User Agents communicate with other User Agents either directly or through intermediate servers. The User Agent also stores and manages call states.

Intermediate servers may behave as proxies or redirect servers. A Proxy Server forwards requests from a User Agent to another SIP server, a User Agent within its network domain, and may collect information for accounting and charging. A Redirect Server responds to client requests and informs of the address of the next server requested. Several hops can be made until the final destination is reached. Basically a server can either maintain the state information or forward request in a stateless manner.

As a general remark, a provisional response (1xx) should be sent as soon as possible when a final response cannot be sent within 200 ms. Algorithms are given in [RFC2543] for how to calculate re-transmission timers for the different SIP messages.

7.4.2 Session Description Protocol

The Session Description Protocol (SDP) is used for describing sessions consisting of audio, video, or multimedia in general. A basic idea was to use it for announcing multicast sessions as described in [RFC2327]. Then session directories may be used to advertise and convey the relevant set-up information to the recipients. SDP, being a format for describing sessions, is independent of the protocol used for carrying this information.

Related to SDP, a few key terms are defined:

- Conference: A multimedia conference is a set of two or more communicating users along with the software they are using to communicate.

- Session: A multimedia session is a set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia conference is an example of a multimedia session.

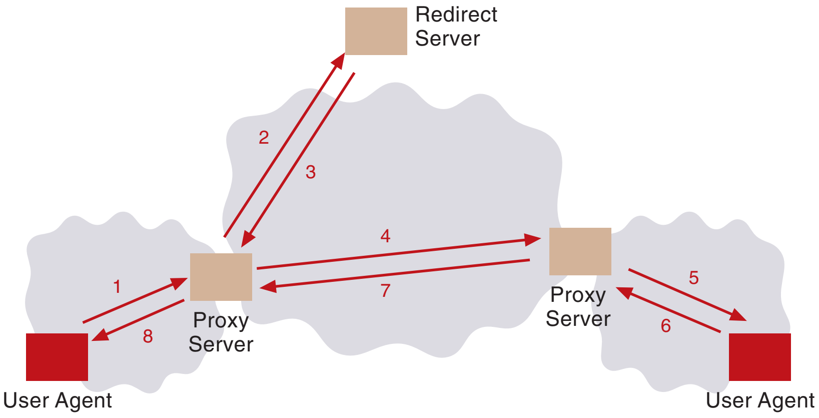


Figure 26 SIP architecture and entities

- Session announcement: A session announcement is a mechanism by which a session description is conveyed to users in a proactive fashion, i.e. the session description was not explicitly requested by the user.

SDP carries the following information:

- Session name and purpose;

- Time(s) the session is active (start/stop times, repeat times);

- Media included in the session (type of media: video, audio, etc., transport protocol, format of the media; H.261, MPEG, etc.);

- Information for receiving/accessing the information (address, port, etc.);

- Information about bandwidth used (e.g. for a conference);

- Contact information for session-responsible person.

The two last ones may be included if found desirable.

The session descriptions are textual, only using ASCII coding. Each line specifies a characteristic in the form: `<type> = <value>`, where `<type>` is one character, see Table 2.

As an example the bandwidth is specified as `b=<modifier>:<bandwidth-value>` where the bandwidth value is given in kbit/s. Two modifiers are possible; either the total bandwidth for all media flows at all sites (called conference total), or bandwidth for a single media flow at a single site (called application-specific maximum).

type	description	type	description
v	protocol version	o	owner/creator and session identifier
s	session name	i	* session information
u	* URI of description	e	* e-mail address
p	* phone number	c	* connection information
b	* bandwidth information	z	* time zone adjustments
k	* encryption key	a	* zero or more session attribute lines
t	time the session is active	r	* zero or more repeat times
m	media name and transport address	i	* media title

Table 2 Type identifiers for SDP (indicates optional type)*

Two examples of attribute lines are the frame rate and a quality indicator:

`a=framerate:<frame rate>`, giving the number of video frame rates per second.

`a=quality:<quality>`, giving a value from 0 to 10 (10 being the best, 5 the default, and 0 the worst but still usable).

7.4.3 Multimedia Conferencing Architecture

The multimedia conferencing architecture elaborated by IETF is depicted in Figure 27.

As shown, SIP may use UDP (unlike HTTP). Then, several SIP messages can be put into the same UDP message. When TCP is used, several SIP transactions can be carried on the same TCP connection. If the server leaves the TCP connection open after returning the reply, the client may use the connection for later SIP messages (or even other protocols, like HTTP).

A similar architecture has also been described for H.323.

7.4.4 Interworking

Supporting the telephony service, more functions must be available, such as servers for handling the service (and supplementary services) as described above. An additional element is a gateway used whenever two domain types are to be traversed. Commonly a gateway is separated into a Media Gateway (MG) and a Media Gateway Controller (MGC) function, see Figure 28. The protocols between MGs could be any of the packet formats for traffic flows. The signalling protocols between the MGCs might be SIP, Bearer Independent Call Control (BICC), or belong to the H.323 family. Megaco is a protocol that may be used between the MG and the MGC.

Similar to SIP, H.323 also defines mechanisms for call routing, call signalling, capability exchange, media control and supplementary services. Work is undergoing to specify interworking between the two protocols, e.g. within the ETSI TIPHON project.

7.4.5 Performance Issues

In order to support transport of real-time traffic flows over an IP network, one must be able to handle:

- Timing and synchronisation of, and between, individual samples of traffic flows for the same applications;
- Effects of packets being lost;
- Effects of packets being delayed;
- Packets arriving in a different order at the receiver than they were sent;
- Multiple traffic flows and different types of traffic flows;
- Monitoring and flow control.

There are multiple ways of implementing the voice transfer service, in the network, but per-

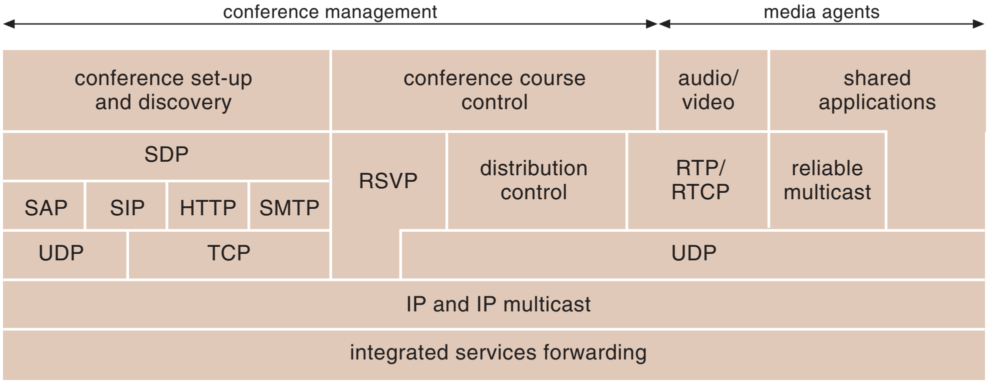


Figure 27 IETF multimedia conferencing architecture

haps even more in the terminal equipment. For the latter, there are factors affecting the user perceived quality, such as (see [Reyn01]):

- Speech coding applied (e.g. G.711, G.726, G.729, G.723.1, GSM);
- Packetisation efficiency, including how many samples are put into the same packet;
- Silence suppression;
- Error-concealment methods;
- Codec-tandem performance.

Examples of these, referring to the E-model, are given in [Reyn01] and [Vlee01].

One may say that the bit rate per voice channel is a measure for efficiency. Basically, this efficiency can be increased by:

- Using low bit rate codings;
- Increasing the packet lengths (less overhead compared to the payload);
- Multiplexing speech samples from several conversations into the same set of packets;
- Compressing headers, e.g. for the combination of IP/UDP/RTP;
- Suppressing silence periods.

8 Concluding Remarks

Facing the dynamic environment, a network operator looks for a general-purpose network. Most often these days, this network is IP-based. However, is should also be "future proof" in the sense that future services are supported. This asks for mechanisms additional to pure IP forwarding, inviting for Traffic Engineering mechanisms. A part of the argumentation, at least to an incumbent network operator, is the need to consolidate his current portfolio of networks, being PSTN/ISDN and others, such as Frame Relay, ATM and X.25. During this, more simple and efficient solutions are sought, including all aspects, like infrastructure, service network, service control, management, etc.

Before the complete network integration has been achieved, interworking between different networks is needed. In particular, the telephony service will likely require gateways between IP-based network and PSTN.

When the access networks are upgraded, for instance by introducing xDSL, the total traffic loads into IP-based networks are expected to increase drastically. This places further pressure on the router networks. The requirements are expected not only to refer to higher throughput measures (e.g. capable of handling several Gbit/s and Tbit/s), but also to offer differentiated and ensured service levels. This allows for supporting a wider range of applications and accompanying ranges of traffic characteristics. However, a lot of challenges remain to be solved for making complete solutions, including customer equipment, service control and management. A suggestion is to logically divide the network into a number of virtual networks, e.g. each of the networks supporting certain types of traffic flow characteristics.

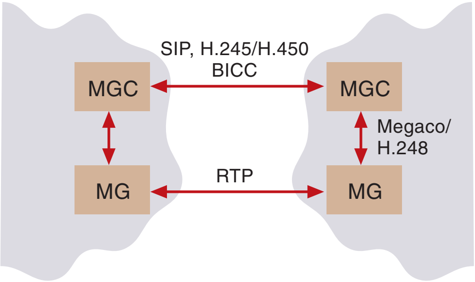


Figure 28 Interconnecting gateways with examples of protocols

Considering the range of customers connected, some more advanced than others, there will also be a need for tailoring the service portfolios. Some customers may well be more or less self-provided (self-service), while others may want more "customer nursing" packages. This again asks for differentiation and adaptation. A particular challenge is to arrive at fast, accurate and efficient ways of implementing such mechanisms in the operator's organisation and systems.

Further supporting mobility implies the need to decouple the static home address of each terminal/user from its current whereabouts. This will place performance/capacity requirements on mobility-like servers. It also means that interoperability/interconnection arrangements between several providers/operators must be solved. Exporting/importing relevant information, e.g. for roaming users, must be done in a secure and efficient way. Other interactions should also be automated, for instance applying e-commerce solutions. This would also open for having more dynamic arrangements between the different actors; customers, network operators and service providers, making the challenges of adequate traffic engineering functions even tougher to fulfil.

In this article the more basic topics have been addressed. These are directed towards the IP-based network, the main protocols, as well as relations with underlying media (fibre) and certain applications of the IP-networks (VPN, telephony, mobility, multicast). By no means is the presentation exhaustive. Simply counting the number of publications being presented every day within the "Internet area" would prevent any complete article from ever being printed.

References

[Come88] Comer, D. *Internetworking with TCP/IP*. Englewood Cliffs, NJ, Prentice-Hall, 1988.

[Feng01] Feng, B et al. State-of-the-art of IP Routing. *Telektronikk*, 97 (2/3), 130–144, 2001. (This issue.)

[ID_BVIPv6] Nguyen, T T et al. 2001. *BGP-MPLS VPN extension for IPv6 VPN over an IPv4 infrastructure.* draft-nguyen-bgp-ipv6-vpn-02.txt. Work in progress.

[ID_GMPLS] IETF. 2001. Ashwood-Smith, P et al. *Generalized MPLS – Signaling Functional Description.* draft-ietf-mpls-generalized-signaling-04.txt. Work in progress.

[ID_IPoptfw] Rajagopalan, B et al. 2001. *IP over Optical Networks: A Framework.* draft-many-ip-optical-framework-03.txt. Work in progress.

[ID_MPLSoptte] Awduche, D O et al. 2001. *Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects.* draft-awduche-mpls-te-optical-03.txt. Work in progress.

[ID_MPmc] Ooms, D et al. 2001. *Framework for IP Multicast in MPLS.* draft-ietf-mpls-multicast-05.txt. Work in progress.

[ID-ppro] Docolsky, D, Bryskin, I. 2000. *Calculating of protection paths and proxy interfaces in optical networks using OSPF.* draft-dovolsky-bryskin-ospf-pathprotect-proxy-00.txt. Work in progress.

[ID_ppvpnfw] IETF. 2001. Callon, R et al. *A Framework for Providing Provisioned Virtual Private Networks.* draft-ietf-ppvpn-framework-00.txt. Work in progress.

[Jens01] Jensen, T. Basic IP-related mechanisms. *Telektronikk*, 97 (2/3), 54–85, 2001. (This issue.)

[Jens01a] Jensen, T. Traffic Engineering – Interdomain and Policy Issues. *Telektronikk*, 97 (2/3), 170–185, 2001. (This issue.)

[Jens01b] Jensen, T. Internet Protocol and transport protocols. *Telektronikk*, 97 (2/3), 20–38, 2001. (This issue.)

[Pain01] Paint, F, Egeland, G. 2001. Seamless Mobility in IP Networks. *Telektronikk*, 97 (1), 83–91.

[Reyn01] Reynolds, R J B, Rix, A W. 2001. Quality VoIP – an engineering challenge. *BT Technology Journal,* 19 (2), 23–32.

[RFC1268] IETF. 1991. Rekhter, Y, Gross, P. *Application of the Border Gateway Protocol in the Internet.* (RFC 1268.)

[RFC1700] IETF. 1994. Reynolds, J, Postel, J. *Assigned Numbers.* (RFC 1700.)

[RFC1771] IETF. 1995. Rekhter, Y, Li, T. *A Border Gateway Protocol 4 (BGP-4).* (RFC 1771.)

[RFC2002] IETF. 1996. Perkins, C (ed.). *IP Mobility Support.* (RFC 2002.)

[RFC2327] IETF. 1998. Handley, H, Jacobson, V. *SDP: Session Description Protocol.* (RFC 2327.)

[RFC2543] IETF. 1999. Handley, M. *SIP: Session Initiation Protocol.* (RFC 2543.)

[RFC2547] IETF. 1999. Rosen, E, Rekhter, Y. *BGP/MPLS VPNs.* (RFC 2547.)

[RFC2764] IETF. 2000. Gleeson, B et al. *A Framework for IP Based Virtual Private Networks.* (RFC 2764.)

[RC2917] IETF. 2000. Muthukrishnan, K, Malis, A. *A Core MPLS IP VPN Architecture.* (RFC 2917.)

[Tane96] Tanenbaum, A S. 1996. *Computer Networks.* Upper Saddle River, NJ, Prentice Hall.

[TS329-3] ETSI. 2001. *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); End-to-End Quality of Service in TIPHON Systems; Part 3: Signalling and Control of end-to-end Quality of Service. V.1.1.1.* (ETSI TS 101 329-3.)

[Vlee01] De Vleeschauwer, D et al. Quality Issues for Packet-based Voice Transport. *Telektronikk*, 97 (2/3), 319–331, 2001. (This issue.)

Voice Transmission over Internet

JOHAN M KARLSSON

The idea of combining both data and voice information in the same network can be traced back to the days of early discussions of ISDN. The envisioned deployment of ISDN was however more technically challenging than planned. The next step into convergence was the introduction of the Internet Protocol (IP). IP was able to route packets through diverse networks without prior knowledge of the service involved. Mostly transfer of files and other text-based information took place. Later, a discussion of differentiated services was introduced, which opened for different kinds of services to be transmitted over the packet switched networks. This paper focuses on the transfer of real-time data and in specific voice over IP (VoIP). Examples of VoIP applications and implementation issues are discussed. The requirements for voice coders and acceptable time delays are mentioned. Further, the specific problems of sending voice samples over a packet switched network as well as some of the protocols supporting this are evaluated.

Johan M Karlsson (40) obtained his MSc and PhD degrees from Lund Institute of Technology, Sweden. He is working on mobile telecommunications and wireless communications issues. Quality of service aspects and performance of traffic and networks, as well as protocol issues are his main research topics. Karlsson is also program director for PCC (Personal Computing and Communications), which is the largest Swedish research initiative in its field.

johan@telecom.lth.se

Introduction

One aspect of real-time transmission stands out as especially important, the use of IP as the foundation for telephony services. The idea is endorsed by many operators around the world. The functionality is often referred to as *Voice over IP* (VoIP) and on more general, non-technical occasions *IP Telephony* or *Internet Telephony*. Although it was designed and optimised to transport data, IP has successfully carried audio and video since its inception. In fact, researchers began to experiment with audio transmission across ARPANET before the Internet of today was in place. By the 1990s, commercial radio stations were sending audio across the Internet, and software was available that allowed an individual to send audio across the Internet or to the standard telephony network. Commercial operators also began using IP technology internationally to carry ordinary telephone calls, i.e. voice.

Network Characteristics

There are mainly two different network categories, namely circuit switched and packet switched networks. The problem with the deployment or integration of voice and data transfer is that they belong to the circuit switched and packet switched category, respectively. To get a better understanding, the two are described below.

Circuit Switching

The traditional networks built for transmitting voice are connection-oriented. This means that to initiate a call the caller has to invoke a certain procedure where a specific route through the entire network is established before any transmission of voice information is carried out. The network signalling system, which is put into action when the caller dials the number of the destination, establishes this route. The route is then used by all information (the pulse code modulated voice) during the call. After finalizing the call the signalling system revokes all the resources that have been allocated on the links and in the switches along the route. The resources (capacity) of the transmission network that are allocated during a call are dedicated entirely for that specific call, i.e. even if nothing is sent at the moment the resources cannot be used for other connections. The capacity allocated is for an ordinary telephone call 64 kb/s. This is due to the fact that the voice is sampled with a frequency of 8 kHz and that each sample is coded by 8 bits, i.e. 64 kb/s. The advantage of the approach to allocate capacity along the route is that no variations in delay between sender and receiver are introduced; the delay that will exist will be fixed and deterministic in its nature.

Packet Switching

The data networks have mainly been built to be connectionless. This means that no connection or resources are allocated for a specific transmission. The information that is to be sent is divided into small segments, called packets, which are sent out on the network independently. All transmissions on such a network have to compete for the available resources in some fair manner. Two different approaches are taken within this concept, datagram and virtual circuit. In the case of datagram the packets from one transmission are sent as if each of them belonged to a new independent transmission. This could very well mean that the packets will be received at the destination in another order than they were transmitted at the sender. For example, after some time the link that seems to be the closest and/or most efficient will be congested or a switch overloaded along the route. The next packet in order to traverse the network will then be routed along a different route in order to avoid this congestion. Due to the new route, this packet could reach the receiver before some of the packets on the congested route.

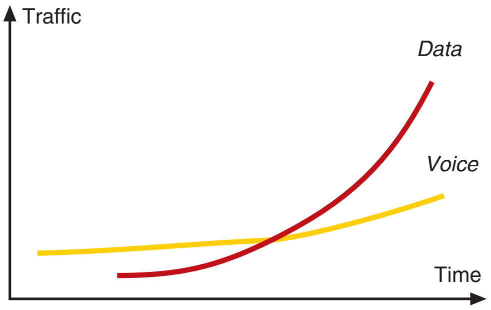


Figure 1 The generic growth of data and voice transmissions in the networks versus time

In the case of virtual circuit a specific route is established for the entire transmission, which all packets belonging to the same transmission will use. In contrast to circuit switched networks, described above, no specific resources are dedicated to this transmission. This means that the packets sent would be received in the same order that they were sent, but the inter-arrival times could differ. The traffic load on the route chosen could vary during the transmission and hence also the inter-arrival times of the packets.

Convergence

In general the connection-oriented networks are thought of as voice networks, while connectionless networks are thought of as data networks. The growth in voice and data transmissions on the networks has had different outcome. A generic diagram of their respective growth is depicted in Figure 1.

The idea of converging both voice and data into the same network is not a new one, however the prerequisite for doing it has not yet been fully present on a more global basis. For many reasons, data networks (i.e. packet switched networks) are generally more efficient than the voice networks (i.e. circuit switched networks). Further, the growth in data traffic is far greater than that of voice traffic. It should however be pointed out that voice connections are still greater in number than data connections. If we look at the convergence of the two networks, it makes much more sense to enhance the capabilities of the packet switched network to also cater for the voice traffic than vice versa. If this is the solution something has to be done to the voice traffic coding and transmission in order to make it suitable for a packet switched environment.

VoIP Applications

The real power of data and telephony integration is its potential to spawn new applications. Integral to this opportunity is the use of industry standard architectures upon which independent third parties can build applications. Such applications have the power to spark radical shifts in collective business behaviour. While the potential impact of this convergence is enormous, the size of the separate installed bases of data and voice make integration a formidable task. Solutions available have generally been limited to point to products with relatively narrow scope and with no answer to the installed base dilemma.

Some examples of voice over IP (VoIP) applications that are possible and also likely to happen are;

- Internet-aware telephones – “ordinary” telephones that are enhanced to also cater for Internet access. An example could be to look up a telephone number in a directory over the Internet and directly dial the received number.

- Voice calls from a mobile PC via the Internet – to use the PC in the hotel room or at a meeting connected to the Internet to call the office for some information. This could also be used to send and receive voice mail.

- Voice calls from Internet cafés – not only to play some games and look at the ordinary email, the Internet café could also be a place for making calls to friends all over the world or receive voice mails.

- Public telephone gateways – this could enable interconnections between the Internet and the public telephone network. Features of this configuration is for example that a voice application on a PC could connect to the public telephone network gateway closest to the destination and from this point the connection goes through the public telephone network. In this way long distance charges are avoided even though the receiver of the call is only connected through the public telephone network.

- Information centre access – through different types of call centres people call to get information. If this voice call is made through the VoIP concept via the Internet the same connection could also contain data transfer. The requested information (long number series, instructions, account statements, pictures, manuals, ...) could be received in parallel. This further enables a continued discussion where both parts have the same document (information) in front of them, which will certainly decrease the possibilities of misunderstandings.

Speech Quality and Characteristics

Speech quality and characteristics is a very important, however subjective matter. The same quality as for ordinary telephony is viewed as a basic requirement, although some experts argue that a cost function versus quality trade-off should be applied.

Delay component	Consumer (objective)	Business (objective)	Today (actual)	Theoretical (minimum)	Above minimum
PC client	*100*	*30*	*150*	*67.5*	*82,5*
Access	*70*	*10*	*150*	*44*	*106*
IP network	*50*	*30*	*96*	*40*	*56*
Gateway/POP	*80*	*30*	*160*	*67.5*	*92,5*
PSTN/phone	*Negligible*	*Negligible*	*Negligible*	*Negligible*	*0*
Total	**300**	**100**	**556**	**159**	**337**

Figure 2 VoIP round-trip delay allocation and current performance in milliseconds (ms), for the different delay components

The following issues have an impact on the QoS obtained by the receiver; delay/latency, jitter, digital sampling, voice compression, digital-to-analogue conversion, tandem encoding, voice activity detection, echo, packet loss and the protocols chosen. Even though the received quality is subjective ITU have tried to make some standardized measures and from these it could be concluded that the three main factors of the impact of the received quality are latency, jitter and packet loss. For latency the two main problems are echo and talker overlap. Echo becomes a problem when the delay increases above about 50 ms, and the talker overlap becomes significant and quite annoying when the delay is 500 ms and more. Both figures apply to the round-trip-delay. The ITU specification G.114 [1] recommends that no more than 300 ms should occur for the connection to be categorized as a high quality connection. (It should be mentioned that the one-way end-to-end delay for transmission to a satellite is approximately 270 ms. Satellite links have been very common and are still widely used for telephone conversations.) For most cases the limit for echo to become annoying will be reached and therefore VoIP have to implement some kind of echo cancellation procedures.

The delay is characterized as the total amount of time it takes to transfer the voice from the sender to the receiver. This time has mainly three components; it is the propagation delay, the serialization delay and the processing delay. A description of the various components of the delay is summarized in Figure 2, where all numbers indicate time in milliseconds. The figures come from a study performed by Bell [2]. The consumer objective (100 ms) and the business objective (300 ms) are assumed values, which are allocated to the six delay components. The PSTN and the telephone client are neither assumed to contribute to the total delay and are therefore combined in the figure and marked “negligible”. The column marked “today” is data from actual measurements performed. The PC Client and Gateway/POP values in the theoretical column may need some explanation. It is assumed that two frames, with an encoding delay of 30 ms each, are captured in one IP packet, and that a look-ahead delay of 7.5 ms is needed.

Packetized Voice

One of the problems of using real-time applications over packet switched networks is that the timing of the sent packets never could be obtained at the receiver end. This is due to the fact that the IP networks are not isochronous, i.e. the exact distance between two packets sent out by the sender will most probably be changed during the time they traverse the networks, since they will experience different conditions on their way to their destination. The conditions within the network and along a route could be modelled according to a probabilistic distribution to be able to determine the expected delay variations. The jitter is usually solved with a buffer; see Figure 3, where the buffer has to be filled with L buffer places before any packet is released to the receiver. In this way, there will hopefully always be a packet to release even though the arrival rate of new packets has decreased momentarily. The arrival rate to the buffer has a probabilistic distribution (λ) while the release from the buffer is done according to a deterministic distribution (d).

The packets traversing the network are not guaranteed in any way to reach their destination, for datagram services not even to reach the destination in order. Individual packets could be lost due to congestion on the links traversed. In normal cases the packet switched networks using IP have retransmission algorithms implemented on higher layers, i.e. the TCP on the transport layer. The retransmission procedures are unfortunately

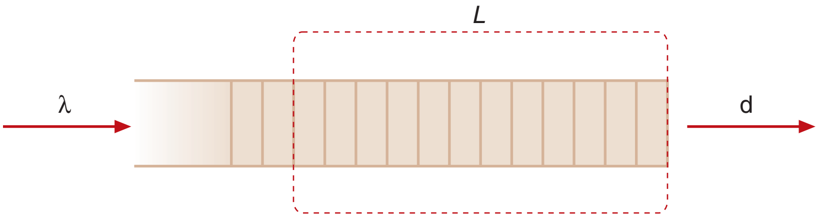


Figure 3 A buffer to compensate for the jitter introduced by the IP networks. L is the number of packets required in the buffer to start releasing packets

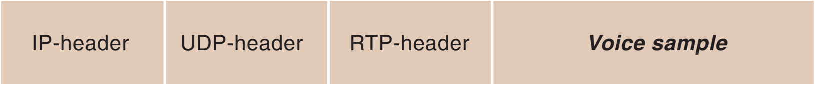


Figure 4 The VoIP packet structure, consisting of the IP, UDP and RTP header in front of the actual voice sample

too slow to cope with the time sensitiveness of real time voice transmissions. This forces VoIP to incorporate other quicker solutions to solve this problem. The implementations used are either to use replay of the last packet, interpolation or introducing redundant information in the packet stream. Due to the redundant structure of our language, these solutions could handle packet losses of at least 5–7 % without major reductions of the perceived quality.

Other ways of improving the speech is to use the silent periods in a call, which could amount to more than half of the time. The silent periods include real silence, pauses made and breaths. No packets are sent with information on “silence”, this interruption of information will be interpreted as total silence at the receiver. To avoid this, the receiver side adds some “nice sounding noise” to the output. Using this approach the required bandwidth could be reduced substantially.

VoIP Packet

The coded voice sample obtained by the voice-receiving unit is encapsulated into the other necessary protocols to be able to be sent on to the network. Firstly, it is encapsulated by the RTP (see section on VoIP Supporting Protocols), then by the UDP and finally by the IP. The UDP (User Datagram Protocol) is a transport protocol, which coexists with TCP (Transmission Control Protocol). For real time applications the UDP is mostly used, while the TCP is more often used for transport of text, sound and pictures in non real time. A generic outlook of the encapsulated packet is shown in Figure 4. A more detailed format of a VoIP packet is shown in Figure 7. As could be seen in the latter figure the 20-octet voice sample becomes at least a 60-octet packet, and many times more than that. (This size is further enlarged before the information enters the network, due to further encapsulation.) The parameters in the headers of the encapsulated VoIP packet format need some explanation. The RTP header includes fields for which version of RTP is used (V), if padding is applied (P), if an extension exists (X), and the number of CSRS identifiers that follow the fixed header (CC). The RTP header also contains a sequence number and time stamp. These two parameters assure that the packets arrive in order and that information is obtained on the actual round trip delay. This is used to calculate the synchronization and to minimize the effects of arrival delay variations (jitter). Finally, the SSRC and CSRC fields are used to identify the different sources, in this case audio sources, which are multiplexed together to create this packet. The following UDP header is self-explanatory.

The version of IP used is outlined in the first header bits, followed by the IHL (IP Header Length), and the type of service field. The latter describes high or low precedence of delay, throughput and reliability of this datagram. The IP header also includes flags which indicate if fragmentation is allowed and if so, how the process should be handled. The fragment offset field indicates where, in the reassembled message, this specific fragment belongs. This is done to be able to reassemble the entire packet in a more efficient way, i.e. to be able to start sorting the fragments before all of them have arrived. The protocol field identifies the higher layer protocol following the IP header, in our case UDP (coded as 17 [3]).

Implementation

The implementation issue is quite complex and several solutions exist to this problem. Basically, there are four options to implement VoIP to an existing network; IP PBX, converged appliances, gateways and other solutions.

The IP PBXs are great for the design of the system and have several features such as being able to manage your phone from your PC, multiline call control and automatic call distribution. Using IP PBX also includes the possibility to create a distributed system throughout an IP network. This means that geographically distributed and separated phones, with features such as direct call, forwarding, conferencing, preset numbers and voice mail, provide the appearance of being connected directly to the local PBX. Using converged appliances which join phone and data networks provide the simplified management that fulfills the promise of VoIP. The user gets voice PBX features with a full complement of data networking, messaging and Internet functions.

The next option is to use a gateway. A VoIP-gateway can be loosely defined as a mechanism that takes circuit switched voice from a traditional PBX, converts it to IP and transfers it across a LAN or WAN to another gateway where it is reconstituted back into a format that is understood by the receiving phone system. Gateway functionality can be obtained through stand-alone boxes, modules or chassis cards for proprietary boxes, also expandable routers of software and expansion cards for some servers. It should however be pointed out that these are voice packets running over IP. But the packets are not running over the Internet, and none of the features and capabilities gained by converging voice and data networks are obtained. Finally, the other option available is mainly integrating the existing voice with IP systems at different points along the chain. The PBX could be IP enabled; only the trunks could be IP-based or maybe just the phones. Technology exists to do everything from single device IP integration to complete infrastructure replacement.

According to a recent survey among VoIP vendors a majority saw a rapid growth in the number of VoIP products that would interoperate within a foreseeable future. The figures given were that by year-end of 2002 72 % of the VoIP products will interoperate, by 2004 88 %, by 2005 94 % and finally by year-end 2005 100 %.

The transition to this new order will likely occur gradually, emerging from organizations back offices and special application workgroups. The current paradigm consists of a circuit switched fabric for voice networks and a complex separate LAN infrastructure for data. The hybrid model, deployed by some enterprises already, is the CTI (Computer Telephony Integration). While most of the data transfer takes place on specific data networks, some have selectively deployed CTI systems for specific applications, generally those designed to generate revenue, such as telemarketing, or minimize costs, such as customer support. In a typical CTI system the incoming caller's telephony number is transferred to the systems database, i.e. computer network, which transforms the customer's telephony number to specific customer information. This information is then displayed on the screen in front of the specialist, sales or support personnel. The connections between the telephony and data networks are loose. The market is severely restrained because it relies on proprietary connections between insular systems. Unlike the world of packet telephony CTI relies on short distance relationships, over proprietary lines, between complementary systems (vendors) that have each been optimized bilaterally for their specific purpose. The final (?) telephony paradigm consists of telephony and data tightly coupled on packet based multimedia networks. In this scenario, data and voice share a common transport network and equipment. Designed with this in mind the fabrics are capable of growing to support new services like video conferencing and video streaming and voice mail. To be able to support all these kinds of services the fabrics have to be equipped with features to cope with streams with different QoS demands. In this scenario the telephony calls become transparent, i.e. the users will not be able to tell whether a call is placed over the packet switched network, the circuit switched network or a combination of the two. This new scenario also has the potential to spark fundamental shifts in collective business behaviour, as people exploit the simultaneous and joined delivery of data applications and voice over a single unified network. This will most likely provide unprecedented opportunities for new enterprises to provide innovative applications.

Voice Coders

Key technical requirements for coders include:

- Low bandwidth (8 kb/s or less);

- High quality for voice (3.5 or higher on the MOS[1] (mean opinion score) scale);

- Low latency.

In real-time transmission, up to 30 % of the packets in a transmission could be lost or delayed to an extent where they have to be calculated as lost. A successful IP telephony application then needs to recover from lost packets by effectively reconstructing the lost data. The complexity of coding algorithms has an impact as well. High complexity increases the cost of the host platform. G.723.1 [5] is emerging as a popular coding choice. It is an algorithm for compressed digital audio over telephone lines. The enduring requirement for coders, however, is that IP telephony systems will be capable of supporting multiple coders and adding more as technology emerges and popularity increases.

Echo Cancellation

VoIP, using ordinary telephones at the endpoints, will cause echo problems and the gateways have to perform some kind of echo cancellation. The ordinary telephone switches do not generally perform any echo cancellation on local lines. The echo is present due to the exchange of information/signals between the two wire and four wire systems. The echo is however not a problem on the local lines, the latency is not long enough to come back as a separate transmission. When using long distance lines echo cancellation is performed within the telephony system. On these lines the time it takes for the signal to propagate back to the sender is long enough to receive a quite disruptive signal. Comparing VoIP to ordinary telephony makes us discover one of the big differences. When VoIP is used, with ordinary telephones at the end stations and an IP network in between, local lines

[1] *For many years the industry has employed a rather subjective scale to determine the quality of a conversation, defined in [4]. This test is based on a number of volunteer testers who listen to a voice sample and grade it according to the following scale; 5:excellent, 4:good, 3:fair, 2:poor and 1:bad.*

of the telephony system are used. Hence, no echo cancellation is performed by the telephony system even though long distance calls are made. That forces VoIP to include echo cancellation among the services supplied.

Voice Activity Detection

In the ordinary telephone network a two-way simultaneous link is set up between sender and receiver. This link carries voice at a rate of 64 kb/s in both directions. Usually only one of the parties is active at any one time; even the active part has breaks and pauses in a normal speech pattern. Hence, the utilization of this two-way link is most of the time less than 40 %. This fact could be used in VoIP to enhance the performance of the transmission and less bandwidth is required to obtain better speech quality using voice activity detection (VAD). A generic outlook of the VAD algorithm is depicted in Figure 5, where it is shown that the algorithm works by detecting the magnitude (dB) and then deciding when the voice is inactive and thereby stopping the transmission of packets in that direction for the moment. To be on the safe side, when cutting the transmission the algorithm waits a fixed amount of time, hang-over time, after it detects a drop in the voice magnitude before it totally stops the voice sample packet transmission. The hang-over time duration is in the magnitude of hundreds of ms (typically 150–250 ms). Another problem is to differ between voice and background noise, and to calibrate itself the VAD is disabled at the beginning of new calls. However, even after that it could be cumbersome to detect when a new voice spurt occurs. The algorithm cut-offs the beginning of each new voice spurt and waits until it is sure that it is a new voice spurt and not, for example, a noise peak. This phenomenon is called front-end speech clipping, and is usually not noticeable for the listener.

Standards

Interoperability among VoIP products has been a major stumbling block to widespread acceptance of the technology. The ITU's H.323 umbrella standard, shown in Figure 6, which was the first posed for VoIP interoperability, proved complex and difficult to implement. As a result, other less-unwieldy standards were posed in its place and until recently, we have seen little consensus on which VoIP standards that would be the most widely implemented. Even though the H.323 standard is the dominating standard at present, most vendors foresee a coexistence of several standards in the arena for quite some time. The most supported standard is H.323 version 2, but version 3 and 4 are rapidly catching up. (It should be pointed out that H.323 version 1 is not forward compatible with the latter standards of H.323.) Other supported standards are SIP (Session Initiated Protocol) by IETF, the Media Gateway Control Protocol (MGCP) and H.248. SIP is an application layer signalling protocol that specifies call control for multiparty sessions, IP phones or multimedia distribution. Unlike H.323, which is based on binary encoding, SIP is a text-based protocol that is usually easier to implement. Further information regarding SIP could be found in [6,7].

MGCP is designed as a simple mechanism to mainly control the gateways. Its function is to control the gateways while relying on external call control intelligence for more complex functions. With the MGCP model, the gateway focuses on the audio signal translation function while a call agent, external to the gateway, handles the signalling and call processing functions. By separating out the internal gateway functions from the external signalling function, the implementation, upgrade and maintenance of the gateway are reduced to a minimum. This increases the likelihood of widespread use of this technol-

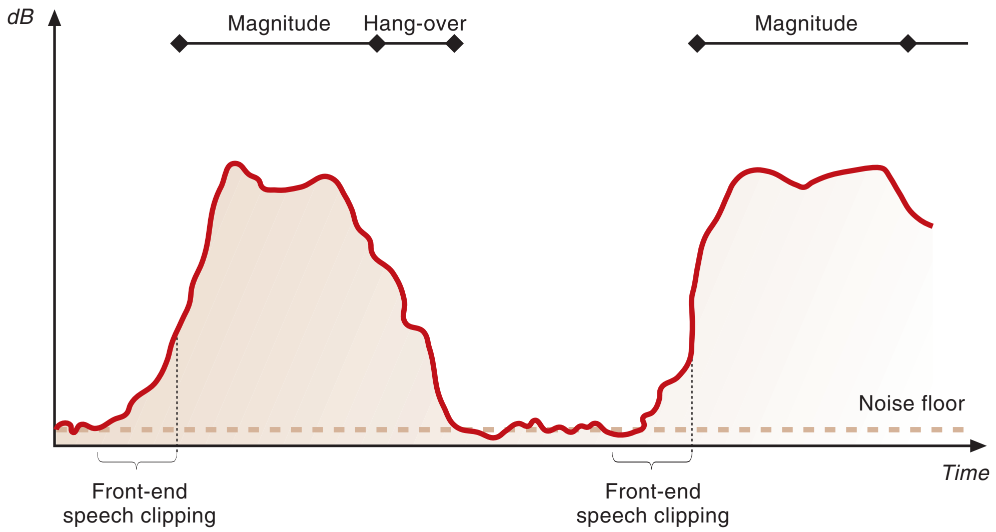


Figure 5 The voice activity detection (VAD) algorithm, used to decrease the required bandwidth for VoIP calls

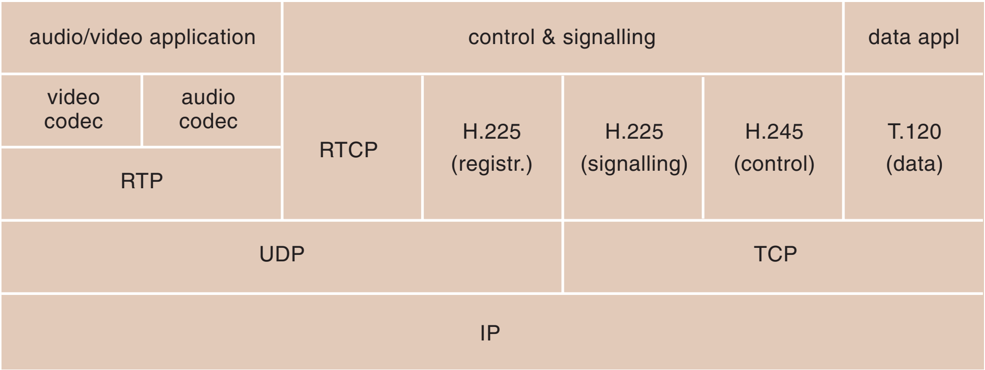


Figure 6 The protocols that comprise the ITU IP telephony standard (H.323)

ogy [8]. The H.248 is a joint venture between IETF and ITU's H.323. The main features are a greater scalability and to address the technical requirements of multimedia conferencing.

There are also some discussion that the technologies will coexist for a long time, however not competing, instead taking their special parts of the system. In such a scenario the H.323 will become the enterprise legacy standard, while MGCP and H.248 will be used between carriers' call agents and other media gateways. The SIP will dominate the connections between the call agents and between call agents and residential IP phones.

Tariff Arbitrage

In today's markets for packet telephony one of the main factors is the tariff arbitrage across the data networks. In most international markets, particularly highly regulated ones, communication carriers have tariff structures that are artificially high as compared to deregulated markets. Additionally, these markets generally offer lower tariff structures for data connections. Several smaller operators have begun to exploit these market disparities and provide users with significant savings on their long distance calls. Internet phones and VoIP gateways (as earlier described) are two products to fully exploit these disparities. Tariff arbitrage products are however purely tactical infrastructure plays, which will be short lived as the international communication carriers embark upon a process of deregulation over the near future. As artificial tariff disparities evaporate, the value proposition will implode. Manufacturers who want to survive this transition must be able to adapt to the changing market condition as they change.

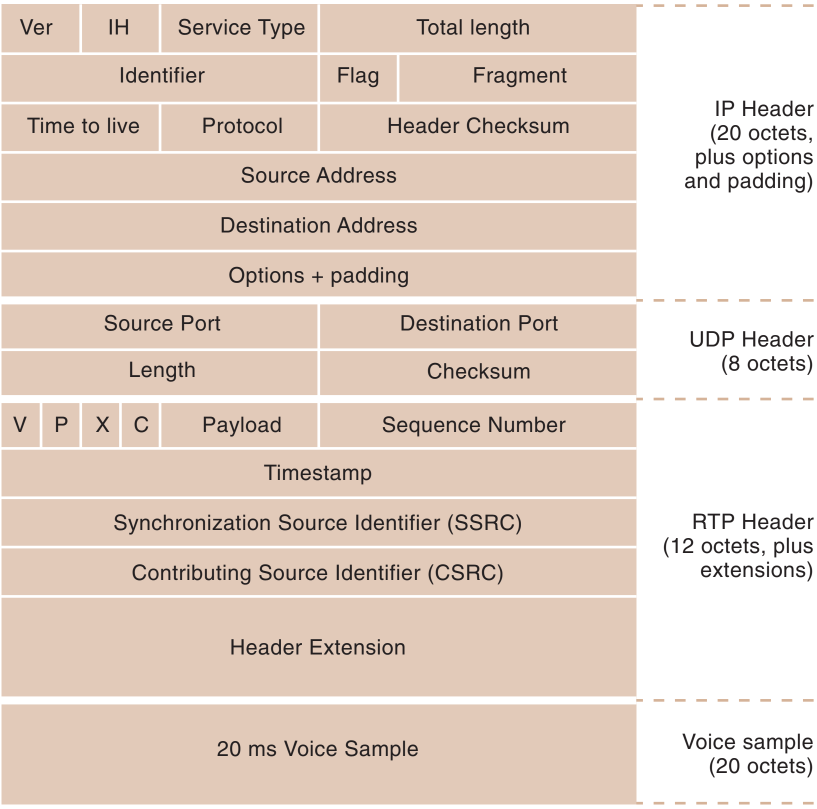
VoIP Supporting Protocols

Most protocols used for delivering the IP packets now also containing voice-coded information were developed with data applications in mind, such as email and file transfer. On higher layers real time protocols are needed to support multimedia applications, such as VoIP. Some examples to be mentioned are;

- IP Multicasting;
- Real-time Transport Protocol (RTP);
- RTP Control Protocol (RTCP);
- Resource Reservation Protocol (RSVP);
- Real-time Streaming Protocol (RTSP) [9].

The objective of *IP Multicasting* is to send one packet and have it received by many destinations. This feature could be used for services like news broadcasts, stock quotes and distance learning. The concept was first introduced in 1989 [10], and involved end terminals have to support the Internet Group Management Protocol (IGMP) [10]. The IGMP enables multicast routers to identify which stations are members of multicast addresses. Specific IP addresses (224.0.0.0 through 239.255.255.255) are reserved to support multicasting [11]. The *Real-time Transport Protocol* provides end-to-end service for data requiring real time support. The IP protocol is deployed

Figure 7 The VoIP packet and its sections. The IP, UDP and RTP headers, which all encapsulate the actual voice sample, are also shown



on a packet switched network, as described earlier. This means that there are no guarantees that the packets will arrive according to the same distribution as they are put on the network, that the packets are received in the right order or that all packets are delivered at all. Applications typically run RTP on top of UDP to make use of the services provided by UDP. The sequence number in RTP is used to reconstruct the sender's packet sequence or to determine a proper location of a packet in a coded packet stream.

The *Real-time Control Protocol* is based on the foundation of its packets being one among others. RTCP packets are regularly inserted into the packet stream and transmitted as ordinary packets. These probe packets are then measured and an estimate of the behaviour of the transmitted service is obtained. As the services introduced gradually become more time sensitive a certain QoS has to be maintained.

The Resource Reservation Protocol is designed to address those requirements. When an application needs a certain QoS grade for its service it consults the RSVP to request support for that level of QoS. The control packets could be sent directly inside IP packets, which are encapsulated by the UDP. One of the drawbacks of this protocol is that to be able to support, provide and promise the requested QoS grade the protocol has to be employed in all routers in the network, or at least all routers along the path of the connection. The RSVP is however not a routing protocol, it is only concerned with the QoS of those packets forwarded by the network's routing protocol. The protocol requests that the receiver and the links along the connection path are reserved to support the data flow.

Finally, the *Real-time Streaming Protocol*, which is an application layer protocol, controls the delivery of the real time packet stream. Examples of services supported by RTSP are to accept additions of media to already existing presentations, as additional media becomes available, and retrievals of media from media servers. The RTSP protocol is in structure rather similar to the HTTP (Hypertext Transfer Protocol), which means that extensions made to HTTP also, in most cases, could be deployed to RTSP. Among the differences between the protocols the out of band data transfer implemented by RTSP should be enlighten.

Conclusions

After the brief introduction of different network switching principles and the concept of sending voice in packets the paper is devoted to VoIP issues. The structure of the VoIP packet as well as some implementation issues were described. The protocols supporting VoIP transmissions, ensuring QoS and other real time and streaming problems were generically described. The requirements for voice coders were also commented on. Even though some problems still exist for VoIP to be an acceptable service with QoS for ordinary telephony, it is slowly being implemented in the networks. It is a very cheap solution for international voice communication, if the lower quality is acceptable. This also indicates that it is, so far, most applicable to direct computer communications. The VoIP protocols are by themselves adequate for a good connection, it is the lower layer protocols, i.e. IP, that need to have some further QoS assurance parameters implemented. The described H.323 family of protocols is designed to provide for interoperability, and several working groups within different standardization organisations are working towards that end.

References

1 ITU. *One-way transmission time.* Geneva, 2000. (ITU-T G.114.)

2 Goodman, B. Internet Telephony and Modem Delay. *IEEE Network Magazine,* 13 (3), 8–16, 1998.

3 Reynolds, J, Postel, J. *Assigned Numbers.* 1994. (RFC 1700.)

4 ITU. *Methods for Subjective Determination of Transmission Quality.* Geneva, 1996. (ITU-T P.800.)

5 ITU. *Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbps.* Geneva, 2000. (ITU-T G.723.1.)

6 Handley, M et al. *SIP: Session Initiated Protocol.* 1999. (RFC 2543.)

7 Schulzrinne, H, Rosenberg, J. The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet. *Bell Labs Technical Journal*, 3 (4), 144–160, 1998.

8 *IETF Media Gateway Control Working Group (MEGACO).* (2001, August 12) [online] – URL: http://www.ietf. org or URL: http://standards.nortelnetworks. com/archives/megaco.html

9 Schulzrinne, H et al. *Real Time Streaming Protocol (RTSP).* 1998. (RFC 2326.)

10 Deering, S. *Host Extensions for IP Multicasting.* 1989. (RFC 1112.)

11 Deering, S, Hinden, R. *Internet Protocol Version 6 (Ipv6) Specification.* 1998. (RFC 2460.)

Quality Issues for Packet-based Voice Transport

DANNY DE VLEESCHAUWER, ANNELIES VAN MOFFAERT, MAARTEN J.C. BÜCHLI, JAN JANSSEN AND GUIDO H. PETIT

This paper studies the influence of the mouth-to-ear delay and distortion (due to voice compression and packet loss) on the quality of a phone call, since these parameters are likely to be larger when the call is transported over a packet-based network instead of over a circuit-switched network. First, the need for echo controlling packetized phone calls is discussed. Second, it is shown that some codecs, in particular predictive codecs, do not attain high enough quality at low bit rates. In the same context also the potential danger of transcoding is recognized. Third, the merit of a packet loss concealment technique to considerably increase the robustness against packet loss is demonstrated. Next, the bounds on the mean one-way mouth-to-ear delay and packet loss that need to be respected in order to attain traditional PSTN quality, are derived for standard codecs (even the recently developed Adaptive MultiRate (AMR) codec) and various levels of echo control (i.e. perfect echo control and standard-compliant echo control). Finally, a gateway-to-gateway scenario in which the transport between the gateways is governed by a Service Level Specification (SLS), is discussed and a numerical example is given to show how the quality bounds can be met in this scenario by tuning the gateway parameters correctly.

Danny De Vleeschauwer (39) is a research engineer participating in the QoS, Traffic and Routing Technology Project within the Network Architecture Team of the Alcatel Network Strategy Group in Antwerp, Belgium.

danny.de_vleeschauwer@alcatel.be

Annelies Van Moffaert (29) is a research engineer participating in the QoS, Traffic and Routing Technology Project within the Network Architecture Team of the Alcatel Network Strategy Group in Antwerp, Belgium.

annelies.van_moffaert@alcatel.be

1 Introduction

The quality of a telephone call depends on the parameter settings in the user terminal and on the parameters of the network over which the call is transported. In this paper, we assume that the user terminals are optimally tuned and study the influence of the network parameters. Offering quality to telephone calls transported over a Public Switched Telephone Network (PSTN) has been understood already for a long time. The main topic of this paper is to investigate how quality can be offered for calls (partly) transported over packet-based networks.

Although currently most of the core of the PSTN is digital, the access parts (e.g. the local loop) are in a lot of cases still analog. There are exceptions however, where even the access is digital, e.g. ISDN access and GSM access. In the 4-to-2-wire hybrids of those analog access parts hybrid echo may be introduced. Additionally, acoustic echo may also be introduced in the user terminals (even when the transport is digital end-to-end). In any case, the level of the echo can be controlled with an echo controller (see ITU-T Recommendation G.168 [3]).

In the PSTN the one-way mouth-to-ear delay mainly consists of propagation delay and switching delay, and hence, it is practically completely determined by the physical distance between both calling parties. An exception is GSM access, where the transport over the air interface alone already introduces about 100 ms of delay [7].

The analog access part of a PSTN is nowadays so short that the distortion introduced in that part of the network is negligible. Over the core of a PSTN the voice signal is (mostly) transported in the G.711 codec format, a format that only introduces a negligible amount of distortion with respect to the analog format. Hence, for most telephone calls transported over a PSTN there is practically no (additional) distortion involved. There are exceptions however, where some distortion is introduced by signal compression: on some transoceanic links the voice is sometimes compressed and in GSM access the voice is transported in a compressed format over the air interface.

When there is little distortion of the voice signal (and when optimally tuned user terminals are utilized), the level of the echo and the one-way mouth-to-ear delay mainly determine the quality of telephone calls transported over a PSTN. It is known that some echo and some delay can be tolerated. ITU-T Recommendations G.114 [1] and G.131 [2] specify the mouth-to-ear delay that can be tolerated (for undistorted voice and) for the case with and without echo control.

The packet-based transport of telephone calls is more flexible than the transport over a PSTN. A packet-based network is not so tightly bound to one codec as the PSTN is to the G.711 codec (which only takes frequencies up to 3.1 kHz into account and has a bit rate of 64 kb/s). Any codec that both user terminals support can be utilized. Wide-band codecs (which take frequencies in the speech signal below 7 kHz into account) could be used to improve the intelligibility of the speech. Note that the bit rate of such a codec is not necessarily higher than the 64 kb/s of the G.711 codec (as the G.711 codec is not very efficient). However, in this paper we only consider low-bit-rate narrow-band codecs, i.e. codecs that like the G.711 only take the frequencies up to 3.1 kHz into account but compress the voice signal to a smaller bit rate than 64 kb/s, possibly at the expense of the introduction of some distortion. On top of this bit rate reduction Voice Activity Detection (VAD) can easily be exploited in packet-based networks, whilst in a PSTN this is impossible.

The price to pay for this additional flexibility is additional complexity: more delay and distortion are likely to be introduced. On top of the delays that also occur in the PSTN, packetization, codec, queuing and dejittering delay come into play [10]. Moreover, the mouth-to-ear delays may considerably differ from one direction to the other, a fact that (practically) never occurs in a PSTN. Distortion may stem from the use of a low-bit-rate codec or from the loss of voice packets in the network or the dejittering buffer. Fortunately, as will be shown in this paper, the one-way mouth-to-ear delay(s) and the distortion can be kept under control by tuning the devices in the network properly.

In the next section we first point out how a packetized phone call differs from a phone call switched over a PSTN as far as quality is concerned. Section 3 quantifies how the echo level, the mouth-to-ear delay(s) and the distortion (through encoding and packet loss) influence the quality of a telephone call by means of the E-model. In Section 4 we present a method to tune the parameters such that adequate quality is attained, when the characteristics with which the voice packets are transported are known, e.g. through a Service Level Specification (SLS). Finally, in the last section we draw the main conclusions.

2 Principles of the Packetized Transport of Phone Calls

As illustrated in Figure 1 there are three essential stages in the packetized transport of phone calls.

In the first stage, the digital voice signal (i.e. a voice signal lowpass-filtered with cut-off frequency at 3.1 kHz that is sampled at 8 kHz and quantized with a linear 13-bit quantizer) is encoded and packetized. This packetization and encoding operation can be performed either in the user terminal or in a gateway. In the latter case we assume that the transport of the voice signal from the user terminal to the gateway (possibly over an analog access network) merely introduces a negligible amount of delay and distortion.

The packetization delay T_{pack} is defined as the time needed to collect all voice samples that end up in one packet, and as such scales linearly with the payload size. The choice of the packetization delay is a trade-off between efficiency (the larger the packets, the smaller the relative influence of overhead bytes) and delay. In fact, the effective bit rate R_{eff} that is needed to transport a voice flow over a packet-based network is defined as

$$R_{eff} = R_{cod} + \frac{S_{OH}}{T_{pack}}, \quad (1)$$

where R_{cod} is the net codec bit rate and S_{OH} the number of overhead bits per voice packet.

Also the encoding performed by a Digital Signal Processor (DSP) needs some time. Besides the voice encoding process other processes run on the DSP as well. An example is an algorithm that detects whether or not the incoming signal is a pure speech signal or consists of (fax, modem

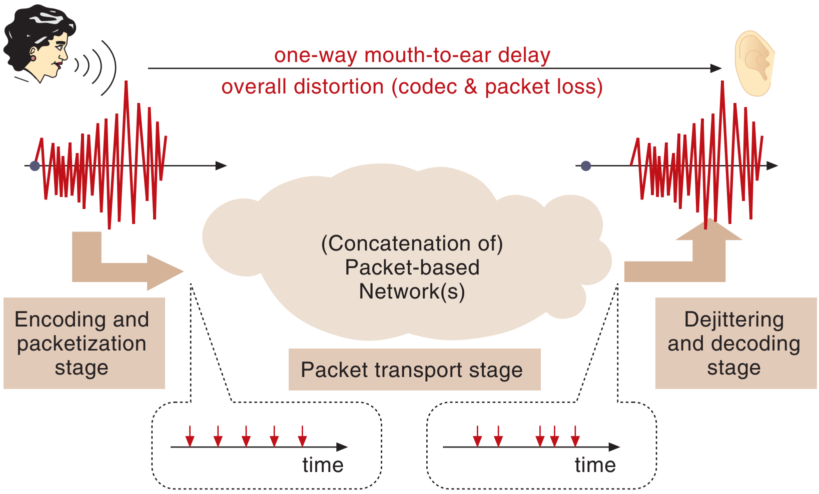

one-way mouth-to-ear delay
overall distortion (codec & packet loss)
(Concatenation of) Packet-based Network(s)
Encoding and packetization stage
Packet transport stage
Dejittering and decoding stage
time
time

Figure 1 Three essential stages in the packetized transport of phone calls

Maarten J.C. Büchli (25) is a research engineer participating in the QoS, Traffic and Routing Technology Project within the Network Architecture Team of the Alcatel Network Strategy Group in Antwerp, Belgium.

maarten.buchli@alcatel.be

Jan Janssen (30) is a research engineer participating in the QoS, Traffic and Routing Technology Project within the Network Architecture Team of the Alcatel Network Strategy Group in Antwerp, Belgium.

jan.janssen@alcatel.be

or DTMF) tones in order to bypass the voice encoder in the latter case. Such an algorithm needs to collect a few samples, as it cannot make an instantaneous decision based on only one sample. This process introduces delay referred to as look-ahead delay. Some encoders themselves already introduce a similar look-ahead delay.

Guido H. Petit (47) is Director of the Network Architecture Team of the Alcatel Network Strategy Group in Antwerp, Belgium.

guido.h.petit@alcatel.be

In the second stage, this flow of packets is transported over a packet-based network consisting of several access and backbone nodes. In the transport of the voice flow over this network some delay is incurred. The network delay can be split into two parts: a deterministic part, referred to as the minimal network delay $T_{net,\min}$, and a stochastic part, referred to as the total queuing delay. The minimal network delay mainly consists of the propagation delay (of 5 μs per km), the sum of all serialization delays, the route look-up delay, etc. If somewhere the packets are transported over an unreliable channel, e.g. an air interface, Forward Error Correction (FEC) techniques, like interleaving coupled with (Reed-Solomon) block or convolutional channel codes, also contribute an amount T_{FEC} to the minimal network delay.

The total queuing delay T_{que} is the sum of the queuing delay in each node. The queuing delay in one network node is due to the competition of several flows for the available resources in the queue of that node. The total queuing delay is responsible for the jitter introduced in the voice flow. The tail distribution function of the total queuing delay is defined as

$$F(T) = \text{Prob}[T_{que} > T]. \quad (2)$$

Note that the inverse of this function evaluated in P, i.e. $F^{-1}(P)$, gives the (1-P)-quantile of the total queuing delay.

In the transport over the network a fraction $P_{loss,net}$ of the packets may get lost. In the case where an unreliable medium (e.g. an air interface) is traversed, a trade-off exists between packet loss in the network and FEC delay introduced in the network

$$P_{loss,net} = G(T_{FEC}). \quad (3)$$

The function $G(.)$ is non-increasing. For reliable channels $G(T_{FEC}) \equiv 0$ and there is no gain in choosing $T_{FEC} > 0$. In this paper we do not consider the transport over an unreliable medium, but refer the interested reader to [14] and [15].

In the last stage the jittered packet flow is dejittered and decoded. Since the decoder needs the packets at a constant rate, dejittering is absolutely necessary. Dejittering a voice flow consists of retaining the fastest packets in the dejittering buffer to allow the slowest ones to catch up. The fastest packets are the ones that do not have to queue in any of the nodes. So, in principle, the fastest packets have to be retained for a time equal to the maximal total queuing delay in the dejittering buffer. Because voice codecs can tolerate some packet loss and because waiting for the slowest packet frequently introduces too much delay, often the fastest packets are retained in the dejittering buffer for a time equal to the (1-P)-quantile of the total queuing delay. This means that a fraction P of the packets will be lost, because they arrive too late. This packet loss introduces distortion. Because it is usually not known if the first arriving packet is a slow or a fast one, a static dejittering mechanism retains the first arriving packet a time T_{jit} in the buffer and then reads the buffer at a constant rate. Dynamic dejittering algorithms are able to gradually learn whether or not the first arriving packet was a fast or a slow one and compensate in that way for the total queuing delay of the first packet.

The decoding and echo control processes finally also introduce some delay.

The dejittering, decoding and echo control can be performed either in the user terminal or in a gateway. In the latter case we assume that the transport of the voice signal from the gateway to the user terminal (possibly over an analog access network) again merely introduces a negligible amount of delay and distortion.

To conclude this section we bring together the impact of all stages on the one-way mouth-to-ear delay T_{M2E} and the overall packet loss P_{loss}.

First, we consider a packetized phone call that is statically dejittered. In that case the one-way mouth-to-ear delay (in one direction) can be split up in the following terms

$$T_{M2E} = T_{pack} + T_{DSP} + T_{net,\min} + T_{que,1} + T_{jit}, \quad (4)$$

where T_{pack} is the packetization delay, T_{DSP} is the sum of encoding, decoding, look-ahead and echo control delays, $T_{net,\min}$ is the total minimal network delay (possibly including the delays over the analog access parts if a gateway is involved and the delay T_{FEC} introduced by the scheme to protect the transport over an unreliable channel), $T_{que,1}$ is the total queuing delay of the first arriving packet and T_{jit} is the dejittering delay. The DSP delay T_{DSP} is lower bounded by the sum of all look-aheads, i.e. even if technology keeps evolving culminating in DSPs with a dazzling processing power, the look-aheads remain unaffected. The minimal network delay $T_{net,\min}$ is lower bounded by the total propagation delay. Since the total queuing delay $T_{que,1}$ of the first packet is stochastic, the one-way mouth-to-ear delay of eq. (4) also is. For static dejittering mechanisms the dejittering delay T_{jit} is usually chosen on the safe side, i.e. such that in the worst case (when the first arriving packet happens to be a fast one) at most a fraction $P_{loss,jit}$ of the packets get lost. Hence,

$$T_{jit} = F^{-1}(P_{loss,jit}) \tag{5}$$

Second, we consider a dynamically dejittered packetized phone call. When the dynamic dejittering mechanism is set to tolerate a packet loss of $P_{loss,jit}$, the dejittering delay is gradually adjusted to compensate for the total queuing delay $T_{que,1}$ of the first packet, so that after a transition period the one-way mouth-to-ear delay tends to

$$T_{M2E} = T_{pack} + T_{DSP} + T_{net,\min} + F^{-1}(P_{loss,jit}). \tag{6}$$

Comparing eq. (6) with eq. (4) combined with (5), we see that adaptive dejittering can (eventually) economize on the one-way mouth-to-ear delay by an amount equal to $T_{que,1}$.

Note that for packetized phone calls the mouth-to-ear delay in one direction is not necessarily the same as that in the reverse direction as each of the terms in eq. (4) (or eq. (6)) may differ from one direction to the other.

Distortion stems from the encoding of the voice signal and from packet loss $P_{loss,net}$ in the transport over the network or from the packet loss $P_{loss,jit}$ in the dejittering buffer, i.e.

$$P_{loss} = 1 - (1 - P_{loss,net})(1 - P_{loss,jit}) \tag{7}$$

Note that also the packet loss (and even the codec format) may differ from one direction to the other.

In the next section we determine how this one-way mouth-to-ear delay and this distortion impact the quality of the call.

3 Parameters Determining the Quality of a Phone Call

3.1 The E-model

The E-model is a tool to predict how an "average user" would rate a phone call of which the characterizing transmission parameters are known. Similar proprietary models exist (see the references in [16]), but the E-model has the advantage that it is standardized in ITU-T Recommendation G.107 [4]. Based on an extensive set of subjective experiments, a scale, referred to as the R-scale, was defined in [8] upon which impairments are approximately additive in the range of interest. Four types of impairments and an advantage factor were identified, that is

$$R = R_0 - I_s - I_d - I_e + A \tag{8}$$

The first term R_0 groups the effects of noise and is amongst other things a function of the level of the circuit noise and the (effective) level of the room noise (present at both sides). The second term I_s includes impairments that occur simultaneously with the voice signal, such as those caused by quantization, by too loud or too soft a connection and by a non-optimum side tone. The third term I_d comprises delayed impairments, including impairments caused by talker and listener echo or by a loss of interactivity. It is mainly a function of the level and the delay of the echo with respect to the original signal and the mouth-to-ear delays in both directions. The fourth term I_e covers impairments caused by what is referred to as "the use of special equipment" in ITU-T Recommendation G.107 and groups effects due to distortion. It is a function of the type of low-bit-rate codec used and the fraction of lost packets. The fifth term A, referred to as the expectation factor, expresses the decrease in rating a user is willing to tolerate because of the "access advantage" that certain systems have over traditional wire-bound telephony. As an example, the expectation factor A for mobile telephony (e.g. GSM) is 10.

Based on the rating R subjective user reactions can be predicted, such as the Mean Opinion Score (MOS) a judging panel would give to the

Table 1 Quality classes according to ITU-T Recommendation G.109

R-value range	90 – 100	80 – 90	70 – 80	60 – 70	0 – 60
Speech transmission quality category	best	high	medium	low	(very) poor

← PSTN quality →

call or the percentage of users finding the quality "Good or Better" (GoB). Moreover, as defined in ITU-T Recommendation G.109 [5] the rating R maps to certain quality classes: a rating R in the ranges [90,100], [80,90], [70,80], [60,70], [50,60] corresponds to "best", "high", "medium", "low" and "poor" quality, respectively. A rating below 50 indicates unacceptable quality. Throughout this paper, the classes are color coded according to Table 1.

In the next paragraphs we study the impact of the one-way mouth-to-ear delay(s) (via I_d) and the distortion (via I_e) on the quality of a packetized phone call. Other factors, like background noise and a connection that is too loud, also impair the quality (via R_0 and I_s) of a packetized phone call, but as these factors are not fundamentally different from a traditional PSTN call, they were not considered. Furthermore, as the objective was to make a fair comparison between the quality of packetized phone calls and traditional wire-bound PSTN calls, the expectation factor A was set to zero.

From Eq. (8) it can be seen that two calls with the same rating R can give a different subjective impression. One call might produce crystal clear, undistorted speech (e.g. $I_e = 0$) but suffer from a relatively large delay (e.g. $I_d = 10$). Another call might slightly distort the speech (e.g. $I_e = 10$), while its delay is not noticeable (e.g. $I_d = 0$). The E-model merely predicts that a judging panel will award the same MOS to both calls and the same percentage of users will find both calls GoB, albeit for different reasons.

Consider a packetized phone call between two parties, referred to as party 1 and party 2 (see Figure 2). Based on the E-model, we evaluate how party 1 will judge the call, that is, what rating R he will assign to it. The influence of delay is studied first, followed by the influence of distortion.

3.2 Influence of Mouth-to-Ear Delay

If there is some delay from party 1 to party 2 and vice versa, the rating R decreases by an amount equal to the impairment I_d. This impairment I_d is the sum of three contributing impairments: impairments due to talker echo, due to listener echo and due to the loss of interactivity. The impairment associated with talker and listener echo depends on the delay and the level of the respective echoes with respect to the original signal. We assume that the echoes (if any) are generated in devices (4-to-2-wire hybrids or user terminals) very close to the calling parties, i.e. that there are no echoes introduced somewhere in (hybrids in) the middle of the network. In that way only the mouth-to-ear delay $T_{M2E,12}$ from party 1 to party 2 and the one $T_{M2E,21}$ from party 2 to party 1 play a role. Remember that in a packet-based environment these two delays may differ.

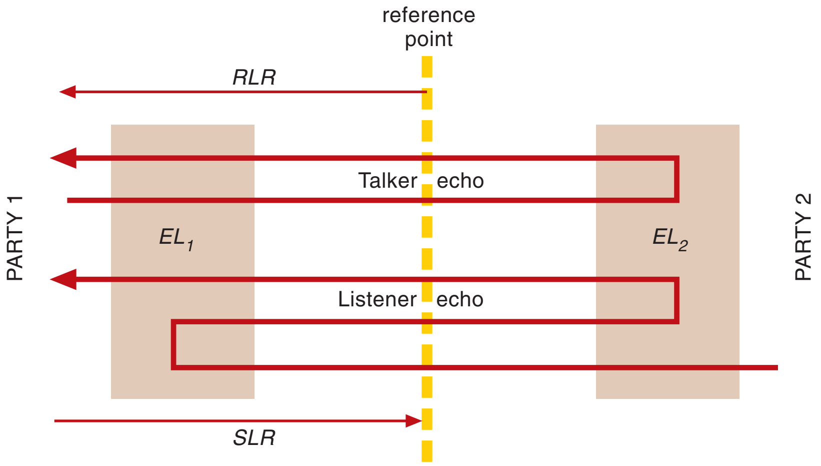


Figure 2 Talker and listener echo

Talker echo disturbs party 1, who hears an attenuated and delayed echo of his own words $T_{M2E,12}+T_{M2E,21}$ after he uttered them. This echo is caused by a reflection close to party 2. This echo is attenuated by $SLR+RLR+EL_2$ (expressed in dB) with respect to the original signal. Here, EL_2 is the echo loss close to party 2 (measured with respect to a certain reference point) [8] and the Send Loudness Rating SLR and Receive Loudness Rating RLR are defined as the attenuation of the signal from party 1 to the reference point and vice versa, respectively. The sum $SLR+RLR$ is usually (tuned to) about 10 dB, a value that we assume in the remainder of this paper.

Second, listener echo also disturbs party 1, who hears the original signal from party 2 followed by an attenuated echo of this signal $T_{M2E,12}+T_{M2E,21}$ after the original signal. The level of this echo is determined by a reflection close to party 1 with attenuation EL_1, followed by a reflection close to party 2 with attenuation EL_2. Hence, the attenuation of the listener echo with respect to the original signal heard by party 1 is EL_1+EL_2 (expressed in dB).

Echo may occur in the hybrid if the packetized phone call is terminated over a local PSTN or in the caller's user terminal. For PSTN calls from traditional handsets, where echo is mainly caused by the 4-to-2-wire hybrids, a typical value for the echo loss is in the order of 20 dB [8]. The same value is valid for packetized phone calls where the call is terminated via a gateway over a local loop to a traditional handset. Acoustic echo is usually small for traditional handsets. It is likely to be higher for other kinds of terminals, such as PCs and handsfree phones

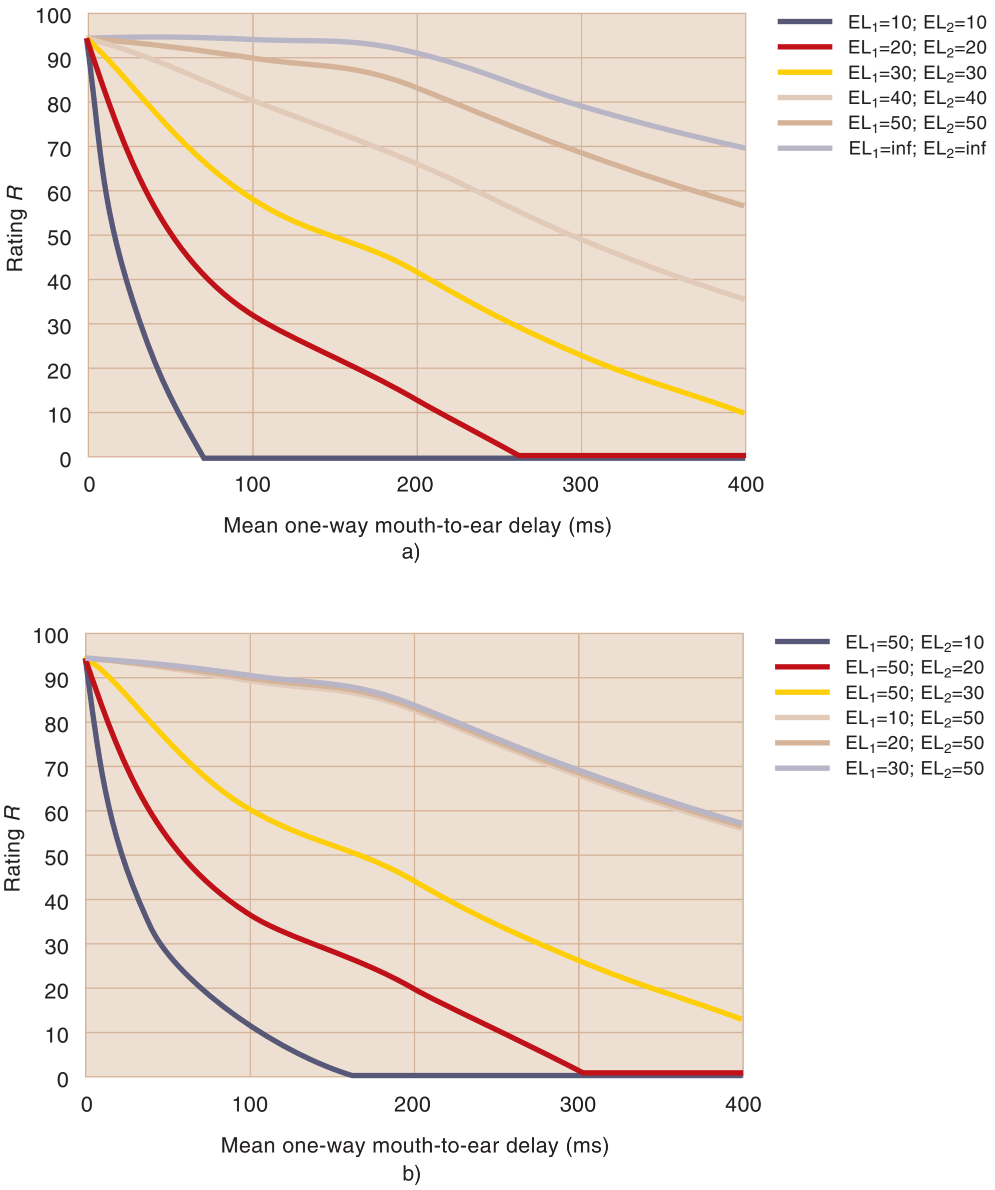


Figure 3 The rating R as a function of the mean one-way mouth-to-ear delay for undistorted voice (i.e. the G.711 codec without packet loss) and for various echo loss values; (a) in case both echo loss values (expressed in dB) are the same, and (b) in case the echo loss values (expressed in dB) are different

(resulting in an echo loss of e.g. 10 dB). An echo controller increases the echo losses EL_1 and EL_2. A standard-compliant echo controller [3] should increase the echo loss by 30 dB. Perfect echo control, which increases the echo losses EL_1 and EL_2 to infinity, can be achieved at moderate computational cost. Since it gradually gets more difficult to control the echo as it is more delayed with respect to its original signal, the echo controller should be deployed as close to the source of echo as possible. Hence, it is recommended that the echo controller in the gateway compensates for the echo generated in the hybrids of the PSTN over which the call is terminated and the echo controller in the terminal compensates for the acoustic echo this terminal generates itself.

The third delay-related factor that may disturb party 1 is the loss of interactivity. If the mouth-to-ear delays are too large, an interactive conversation becomes impossible. The impairment associated with the loss of interactivity is a function of the sum of both mouth-to-ear delays $T_{M2E,12}$+$T_{M2E,21}$.

Hence, under the above mentioned assumptions the impairment I_d is a function of $T_{M2E,m}$, EL_1 and EL_2, with

$$T_{M2E,m} = \frac{T_{M2E,12} + T_{M2E,21}}{2} \qquad (9)$$

the mean one-way mouth-to-ear delay. Figure 3 illustrates the behavior of this function. Figure 3(a) shows how the rating R drops (due to an increase in I_d) as the mouth-to-ear delay increases for different values of the echo loss for the case when the echo losses at both end points are equal ($EL_1 = EL_2$) and when there is no distortion, i.e. $I_e = 0$. The impairment associated with delay is strongly influenced by this echo loss value. Note that the rating R is a non-increasing function of the mouth-to-ear delay. The intrinsic quality of a phone call is defined as the rating R associated with a zero mouth-to-ear delay. The intrinsic quality of a packetized phone call transported without packet loss in the G.711 format and with all other parameters optimally tuned, corresponds to a rating R of about 94. This rating is referred to as $R_{\text{int},G.711}$. Figure 3(a) shows that if echo is perfectly controlled ($EL_1 = EL_2 = \infty$), the phone call retains its intrinsic quality up to a mean one-way mouth-to-ear delay of about 150 ms.

ITU-T Recommendations G.114 [1] and G.131 [2] specify the following tolerable mouth-to-ear delays for traditional PSTN calls:

- Under normal circumstances (i.e. if the echo loss is at least 20 dB), echo control is needed if the mouth-to-ear delay is larger than 25 ms;

- When the echo is adequately controlled:

  - a mouth-to-ear delay of up to 150 ms is acceptable for most user applications;

  - a mouth-to-ear delay between 150 ms and 400 ms is acceptable, provided that one is aware of the impact of delay on the quality of the user applications; and

  - a mouth-to-ear delay above 400 ms is unacceptable.

It can be seen from Figure 3(a) that for an echo loss of 20 dB, the rating R drops below 70 at a mouth-to-ear delay of 25 ms and for calls with perfect echo control, the rating R drops below 70 at a mouth-to-ear delay of 400 ms. Hence, ITU-T Recommendations G.114 and G.131 ensure that traditional PSTN calls have a rating R of at least 70. Also, the interactivity bound of 150 ms can be observed in Figure 3(a) for infinite echo loss.

Figure 3(b) shows how party 1 rates the call in case the echo losses at both end points are different. It can be seen that party 1 experiences a low quality if the echo loss EL_2 close to party 2 is not high enough, even if the echo controller close to party 1 (i.e. his "own" echo controller) is standard-compliant. Alternatively, if the echo controller close to part 2 is good enough, the echo controller close to party 1 does not impact the quality experienced by party 1 a great deal. Hence, the party with the best echo control will experience the worst quality (if all other factors are equal for both parties).

3.3 Influence of Distortion

If the voice signal party 1 hears is distorted, the rating R decreases by an amount equal to the distortion impairment I_e. This impairment is a function of (at least) two parameters: the codec used by party 2 to encode the voice signal and packet loss P_{loss} during the transport of voice packets from party 2 to party 1. Note that it is common practice, but not strictly mandatory, to transport the voice in the same format in both directions.

We first consider the influence of compressing the voice signal. As the G.711 codec just samples the (low-pass filtered) voice signal at 8 kHz and quantizes the samples with a non-uniform logarithm-like 8-bit quantizer, it introduces hardly any distortion. The packetization delay can be any multiple of 0.125 ms.

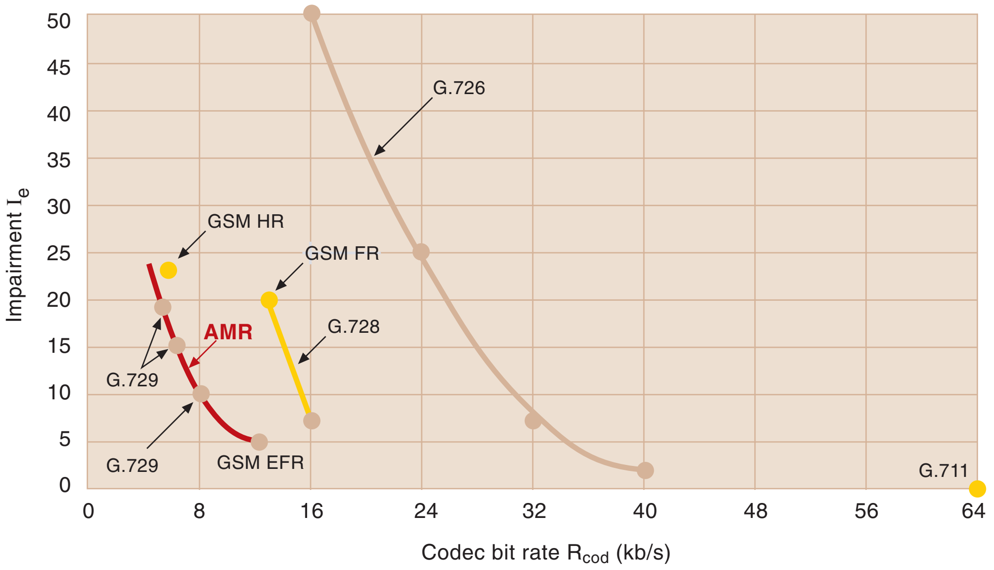
Predictive codecs (e.g. the G.726 codec) predict the sample to be encoded based on the previous ones (already encoded) and quantize the prediction error in 2, 3, 4 or 5 bits, resulting in a net codec bit rate R_{cod} of 16, 24, 32 and 40 kb/s respectively. Again the packetization delay can be any multiple of 0.125 ms.

Codecs of the vocoder type are based on a model for the human vocal track. These codecs first segment the speech signal in intervals of constant duration (referred to as voice frames). Then for each consecutive voice frame, they estimate and quantize the parameters of the vocal track model and collect all quantized parameters in a code word. The net codec bit rate R_{cod} is the code word size (in bits) divided by the frame length. Some of these codecs require a look-ahead in order to estimate the vocal track model parameters more accurately. Since the packetization delay is an integer multiple of the voice frame, and hence is at least one voice frame, the larger the voice frame is, the larger is the minimal delay the codec introduces. Most vocoder codecs have a frame length between 10 and 30 ms (the G.729 codec has 10 ms, the G.723.1 codec 30 ms and all GSM codecs 20 ms). An exception is the G.728 codec, which has a voice frame length of 0.625 ms.

Recently a new codec, the Adaptive MultiRate (AMR) codec [9], was developed in the framework of the third generation mobile network. It has a voice frame length of 20 ms (as all GSM codecs) and the particularity that the vocal track parameters can be quantized in a different number of bits, resulting in code words of variable size, from voice frame to voice frame, and hence, in a variable bit rate.

Figure 4 summarizes the distortion impairment associated with some standardized codecs. The points on this figure are rate-distortion pairs determined by experiments reported in [6]. Also three lines connecting similar pairs are drawn on this figure. This is a straight line when there are

Figure 4 Impairment I_e for several standardized codecs. The points are experimental rate-distortion pairs. Based on this experimental input the rate-distortion (codec bit rate R_{cod} vs. Impairment I_e) curve is interpolated for the G.726 codec, the G.728 codec and the AMR codec



just two pairs or a quadratic best fitting curve in case there are more pairs. One line is associated with the G.726 codec and gives the rate-distortion trade-off for predictive codecs. It can be seen that at low bit rates predictive codecs introduce a lot of distortion. Another line is associated with the G.728 codec. This codec has a better rate-distortion trade-off than predictive codecs but does not reach the full potential of codecs of the vocoder type, as its voice frame size is too small. Also the older GSM-FR and GSM-HR codecs do not reach the full potential of vocoder codecs. A third line is drawn through the state-of-the-art codecs of the vocoder type (i.e. the G.729, G.723.1 and GSM-EFR codec) and as such gives the rate-distortion trade-off for vocoder codecs. It can be seen that vocoder codecs have the best rate-distortion trade-off. Although the AMR codec has not been characterized yet in terms of how much distortion it introduces at what bit rate, the latter curve on Figure 4 (labeled "AMR") forms a very good initial estimate.

A VAD scheme, which detects if the signal contains active speech or background noise, can be used to further reduce the overall bit rate to be sent. Good VAD schemes hardly introduce any additional distortion.

The distortion impairment I_e associated with a codec increases as the packet loss ratio increases. Only a few results are known and are summarized in Figure 5. In that figure we draw the quadratic curves that best fit the experimental data (i.e. the points in that figure) reported in [6], which gives experimental data for four codecs under the assumption that voice packets are lost at random. Although other results are not yet known some trends can be observed.

The sensitivity to packet loss depends on the Packet Loss Concealment (PLC) technique used by the codec. In contrast to the G.711 codec, most state-of-the-art low-bit-rate codecs (e.g. G.729, G.723.1 and GSM-EFR) have a built-in PLC scheme. However, a (proprietary) PLC

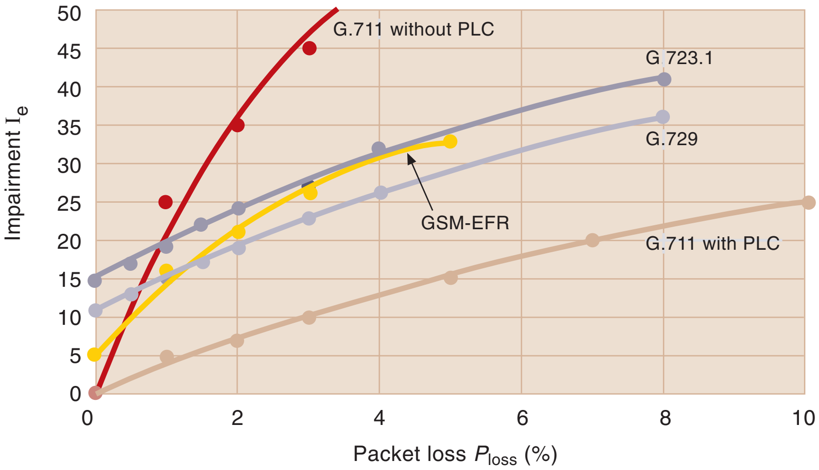


Figure 5 Distortion impairment I_e as a function of the packet loss P_{loss}

CODEC	G.711 (64kb/s)	G.726 (40kb/s)	G.726 (32kb/s)	G.726 (24kb/s)	G.726 (16kb/s)	G.728 (16kb/s)	GSM-FR (13kb/s)	G.728 (12.8kb/s)	GSM-EFR (12.2kb/s)	G.729 (8kb/s)	G.723.1 (6.3kb/s)	GSM-HR (5.6kb/s)	G.723.1 (5.3kb/s)
G.711 (64kb/s)	94	92	87	69	44	87	74	74	89	84	79	71	75
G.726 (40kb/s)	92	90	85	67	42	85	72	72	87	82	77	69	73
G.726 (32kb/s)	87	85	80	62	37	80	67	67	82	77	72	64	68
G.726 (24kb/s)	69	67	62	44	19	62	49	49	64	59	54	46	50
G.726 (16kb/s)	44	42	37	19	0	37	24	24	39	34	29	21	25
G.728 (16kb/s)	87	85	80	62	37	80	67	67	82	77	72	64	68
GSM-FR (13kb/s)	74	72	67	49	24	67	54	54	69	64	59	51	55
G.728 (12.8kb/s)	74	72	67	49	24	67	54	54	69	64	59	51	55
GSM-EFR (12.2kb/s)	89	87	82	64	39	82	69	69	84	79	74	66	70
G.729 (8kb/s)	84	82	77	59	34	77	64	64	79	74	69	61	65
G.723.1 (6.3kb/s)	79	77	72	54	29	72	59	59	74	69	64	56	60
GSM-HR (5.6kb/s)	71	69	64	46	21	64	51	51	66	61	56	48	52
G.723.1 (5.3kb/s)	75	73	68	50	25	68	55	55	70	65	60	52	56

Table 2 Transcoding matrix

scheme can be implemented on top of the G.711 codec. From Figure 5 it can be seen that for the codecs that use PLC, the impairment increases by about 4 units on the *R*-scale per percent packet loss (for low loss values). If no PLC scheme is implemented on top of the G.711 codec, the distortion impairment increases by 25 units on the R-scale for each percent packet loss (for low loss values).

Figure 5 deals only with one specific packetization interval per codec (10 ms for G.711, 20 ms for G.729 and GSM-EFR, 30 ms for G.723.1). The G.723.1 codec was only used at 6.3 kb/s. Comparing the slopes of the curves in Figure 5, we see that the G.711 codec with PLC is slightly less sensitive to packet loss than the G.729 codec, which in turn is a bit less sensitive than the G.723.1 codec. From the results it cannot be concluded if this is due to the bit rate of the codecs (high-bit-rate codec formats contain more redundant information, and hence are probably less sensitive to loss) or to a smaller packetization interval. Also from Figure 5 it can be seen that the PLC technique of the GSM-EFR codec does not perform so well as the PLC techniques of the other considered codecs. The conclusion from this paragraph is that in lossy environments a PLC is highly recommended.

The voice signal does not need to be transported in the same format end-to-end. Somewhere along the route, the voice signal might be transcoded from one codec format into another. Since all (considered) standard codecs need an 8 kHz stream of uniformly quantized voice samples at the input, the code words of the first codec need to be decoded before the signals can be encoded into another codec format. Consequently, the impairment terms associated with the two codecs should be added to obtain the overall distortion impairment I_e, because in the E-model, impairments are approximately additive on the *R*-scale. The intrinsic quality associated with all combinations of two codecs can be found in Table 2 (using the color code of Table 1). The diagonal entries in this table correspond to tandeming two codecs of the same type. Table 2 readily shows that transcoding can be very harmful to the quality of a call. In practice, the order in which the codecs are tandemed has a small influence, which cannot be seen in (the symmetric) Table 2 because, as impairments are considered to be additive in the E-model, asymmetries cannot occur. The conclusion from Table 2 is that transcoding should be avoided.

Standard body	Short name	Codec bit rate (kb/s)	T_{M2E} (ms) EL = infinite	T_{M2E} (ms) EL = 50 dB
ETSI	GSM-HR	5.6	177	29
ITU-T	GSM-FR	13	21	106
ITU-T	G.711	64	400	291
ITU-T	G.728	12.8	210	106
		16	322	243
ITU-T	G.726 G.727	16	NA	NA
		24	NA	NA
		32	322	243
		40	375	276
ITU-T	G.723.1	5.3	219	131
		6.3	251	187
ITU-T	G.729	8	294	223
ETSI	GSM-EFR	12.2	342	256
3GPP	AMR	4.75	197	72
		5.15	214	117
		5.9	239	172
		6.7	262	198
		7.4	280	213
		7.95	293	223
		10.2	332	250
		12.2	342	256

Table 3 Tolerable mean one-way mouth-to-ear delay T_{M2E} bounds when there is no packet loss in the case of perfect echo control (EL=inf) and an echo loss EL = 50 dB. (NA = Traditional PSTN quality (R = 70) is Not Attainable.)

Standard body	Short name	Codec bit rate (kb/s)	P_{loss} EL = infinite T_{M2E} < 150 ms	P_{loss} EL = 50 dB T_{M2E} = 150 ms
ITU-T	G.711 no PLC	64	1.2	0.9
ITU-T	G.711 PLC	6.4	9.6	6.1
ITU-T	G.723.1	6.3	2.1	0.6
ITU-T	G.729	8	3.5	1.7
ETSI	GSM-EFR	12.2	2.5	1.5
3GPP	AMR	4.75	0.7	NA
		5.15	1.2	NA
		5.9	1.9	0.3
		6.7	2.6	1.1
		7.4	3.2	1.6
		7.95	3.5	2.0
		10.2	4.6	3.0
		12.2	4.8	3.2

Table 4 Tolerable packet loss P_{loss} bounds for a mean one-way mouth-to-ear delay below 150 ms in the case of perfect echo control and for a mean one-way mouth-to-ear delay of 150 ms for an echo loss EL = 50 dB. (NA = Traditional PSTN quality (R = 70) is Not Attainable.)

4 Controlling Voice Quality

The conclusion from Section 3 is that for our purposes the rating R can be written as

$$R = R_{\mathrm{int},G.711} - I_d(T_{M2E,m}, EL_1, EL_2) - I_e\,(\mathrm{codec}, P_{loss}) \qquad (10)$$

The combined effect of the first and second term is illustrated in Figure 3. The third term is displayed in Figure 5.

4.1 Quality Bounds

Since the echo control bound of 25 ms is almost always exceeded when the phone calls are transported over a packet-based network, echo control is strongly recommended for packetized phone calls. A good echo controller, i.e. an echo canceller compliant with ITU-T recommendation G.168 [3], can increase the echo loss (from 20 dB usually occurring in the PSTN) to 50 dB. With an echo controller with a non-linear element perfect echo control, in which case the echo loss is increased to infinity, can be achieved.

From Figure 3 it is clear that in the case of perfect echo control at both sides, the intrinsic quality of the call is attained if the mean one-way mouth-to-ear delay is kept below 150 ms. From eq. (10) we notice that this intrinsic quality is solely determined by the distortion impairment I_e, which in turn is determined by the codec(s) used and the overall packet loss experienced. Since the intrinsic quality $R_{\mathrm{int},G.711}$ of an undistorted call is about 94 and the bound for traditional quality is 70, there is an impairment budget of 24, part of which is consumed by the codec(s) (see Figure 4). Once the codec has been chosen, the remainder of the margin can be consumed either by allowing the mean one-way mouth-to-ear delay to exceed 150 ms or by tolerating some packet loss. The bound on the mean one-way mouth-to-ear delay for a certain codec is derived by subtracting the impairment associated with that codec (displayed in Figure 4) from the curves of Figure 3 and determining where the curve associated with perfect echo control drops below 70. The bound on packet loss for a certain codec is derived by determining in Figure 5 for which packet loss value the impairment budget of 24 is just not consumed. The bounds for the AMR codec are derived under the assumption that the interpolation (i.e. the curve in Figure 4 labeled "AMR") is valid and that per percent packet loss 4 units are added to the impairment I_e. The fourth column of Table 3 and Table 4 gives the codec-dependent bounds on the mean one-way mouth-to-ear delay and packet loss, respectively, when the echo is perfectly controlled [10].

The fifth column of the same tables shows how these bounds reduce when the echo control is not perfect, but still very good *EL* = 50 dB (i.e. compliant with ITU-T Recommendation G.168 [3]) at both sides. For the packet loss bounds it was assumed that the mean one-way mouth-to-ear delay was exactly 150 ms. The bounds for the AMR codec are derived under the same assumption specified above. It can be seen that if the performance of the echo controller drops from perfect to slightly less than perfect, this can have a drastic effect, especially on the mean one-way mouth-to-ear delay bound.

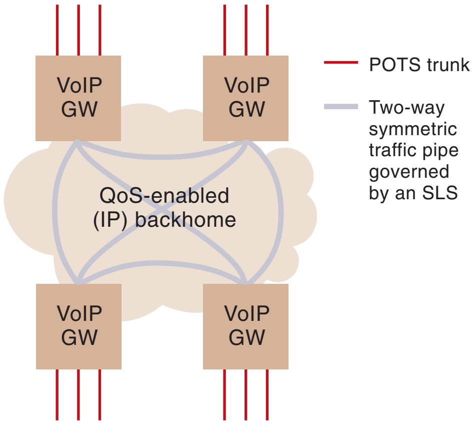

VoIP GW
VoIP GW
QoS-enabled (IP) backhome
VoIP GW
VoIP GW
POTS trunk
Two-way symmetric traffic pipe governed by an SLS

Figure 6 A gateway-to-gateway scenario to transport phone calls

4.2 Controlling the Delay and Distortion in a Gateway-to-Gateway Scenario

In this section we consider a gateway-to-gateway scenario illustrated in Figure 6. Phone calls originate from and terminate at traditional telephone sets and are switched over a local PSTN to gateways, between which the voice signals are transported over a QoS-enabled IP backbone administered by one network manager [12]. Between each pair of gateways a traffic pipe is defined. We assume symmetric pipes. The transport of the voice packets over this pipe is governed by a Service Level Specification (SLS) [13]. The SLS is completely defined by specifying values for $P_{loss,net}$, $T_{net,\min}$ and enough information to describe the function $F(.)$ of eq. (2) as accurately as possible. As described above the latter requirement boils down to specifying as much quantiles of the total queuing delay as necessary.

The gateway parameters that can be tuned are the packetization delay T_{pack} and the dejittering delay T_{jit} (or equivalently the dejittering loss $P_{loss,jit}$ (see eq. (5)). We assume that adaptive dejittering is used and is converged to its optimal value, and hence, eq. (6) determines the one-way mouth-to-ear delay. We furthermore assume that there is no packet loss in the backbone $P_{loss,net}$ = 0 and that the minimum network delay $T_{net,\min}$ is primarily determined by propagation (of 5 µs per km). Hence, this delay $T_{net,\min}$ is determined once the physical distance between the gateways is known.

The choice in packetization delay T_{pack} is a trade-off between efficiency (see eq. (1)) and delay (see eq. (6)). We know from the previous section that under perfect echo control at both

SLS specification	
P	(1-*P*) quantile (ms)
1.E-01	1
1.E-02	3
1.E-03	10
1.E-04	30
1.E-05	130

a)

T_{pack} (ms)	10	20	30	40
R_{eff} (kb/s)	40.0	24.0	18.7	16.0

b)

T_{pack} (ms) / $P_{loss,jit}$	10	20	30	40
1.E-01	52	52	51	51
1.E-02	76	75	75	75
1.E-03	79	79	79	78
1.E-04	79	79	78	78
1.E-05	72	70	69	67

c)

Table 5 (a) SLS specification, (b) effective codec rate R_{eff} (kb/s), and (c) rating R for various values of the packetization delay and dejittering loss sides a total budget of 150 ms can be consumed without hampering the quality. However, from Table 3 it can be seen that if the performance of the echo control reduces to "nearly perfect", but still is standard-compliant, the bound on the mean one-way mouth-to-ear delay can be below 150 ms in some cases. The packetization delay is typically chosen between 10 and 80 ms. From eq. (1) it follows that since the overhead S_{OH} is 320 bits (consisting of 20 IP, 8 UDP and 12 RTP bytes) for Voice over IP (VoIP), an overhead bit rate between 32 kb/s and 4 kb/s, respectively, is introduced.

The flexibility in the choice in dejittering loss $P_{loss,jit}$ (or equivalently the dejittering delay T_{jit}) is governed by the number of quantiles that are specified in the SLS, i.e. how many points of the function $F(.)$ are given. If only the maximum total queuing delay (i.e. the (1-P)-quantile with $P = 0$) is given, only this total queuing delay can be used as dejittering delay. The more quantiles are specified, the more flexible the choice can be.

To conclude this section we give an example. Consider a phone call from Europe to the US. We assume $T_{net,\min} = 50$ ms, $P_{loss,net} = 0$ and that the SLS in both directions is as described in Table 5(a). Furthermore, we assume a DSP delay $T_{DSP} = 15$ ms, an echo loss $EL_1 = EL_2 = 50$ dB, and that the G.729 codec (at 8 kb/s) is used. Table 5(b) gives the effective bit rate R_{eff} calculated with eq. (1) and Table 5(c) (using the color code of Table 1) gives the rating R calculated with eq. (10) for various values of the packetization delay and dejittering loss. From these tables it can be concluded that a packetization delay of 30 ms and a dejittering loss of 10^{-3} lead to a good compromise between effective bit rate ($R_{eff} = 18.7$ kb/s) and quality ($R = 79$).

The question how to provision the SLSs, i.e. how to configure the routers in the network such that the quantiles specified in the SLS are attained is beyond the scope of this paper. We refer the interested reader to [11] and [17].

5 Conclusions

In this paper the quality issues associated with the packetized transport of phone calls were considered. Since for packetized phone calls more delay and distortion is introduced than for traditional PSTN calls, the impact of delay and distortion on the quality of the phone call was studied quantitatively with the E-model. The trade-offs involved in the choice of the packetization delay and dejittering loss were discussed. From this quality study the following conclusions were drawn.

For packetized phone calls echo control is highly recommended, if not required, since otherwise the tolerable mouth-to-ear delay budget risks being too small. If the echo is perfectly controlled, the quality remains equal to the intrinsic quality up to a mouth-to-ear delay of about 150 ms. The intrinsic quality depends on the amount of distortion that is introduced. If the echo control is slightly less than perfect, but still standard-compliant, the quality decreases even for delays smaller than 150 ms.

The intrinsic quality associated with predictive codecs at low bit rates is lower than the traditional PSTN quality. Therefore, these codecs should not be used at a bit rate below 32 kb/s. For the same reason, transcoding should be avoided.

Under perfect echo control the margin between the intrinsic quality of a codec and the bound for traditional quality can either be consumed by allowing a mouth-to-ear delay above 150 ms or by allowing some packet loss. The maximum tolerable bounds on the mean one-way mouth-to-ear delay and packet loss are reported in this paper for the most common codecs and even the recently developed Adaptive MultiRate (AMR) codec. It is also shown how these bounds decrease if the echo control is slightly less than perfect, but still standard-compliant.

These tolerable bounds should be respected by any packetized phone call (gateway-to-gateway, IP-phone-to-IP-phone, mobile-phone-to-mobile-phone, gateway-to-IP-phone, etc.) if traditional quality is to be maintained.

Finally, to illustrate how these bounds can be used this paper considered a gateway-to-gateway scenario where the transport of the voice packets is governed by a Service Level Specification (SLS). The trade-offs involved were shown by means of a numerical example.

Acknowledgement

This work was carried out within the framework of the project LIMSON sponsored by the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT).

References

1 ITU. *One-Way Transmission Time*. February 1996. (ITU-T G.114.)

2 ITU. *Control of Talker Echo*. August 1996. (ITU-T G.131.)

3 ITU. *Digital Network Echo Cancellers*. April 1997. (ITU-T G.168.)

4 ITU. *The E-model, a Computational Model for Use in Transmission Planning.* December 1998. (ITU-T G.107.)

5 ITU. *Definition of Categories of Speech Transmission Quality.* September 1999. (ITU-T G.109.)

6 ITU. *Provisional Planning Values for the Equipment Impairment Factor I_e.* September 1999. (Appendix to ITU-T G.113 (Draft).)

7 ETSI. *Digital Cellular Telecommunications System; Technical Performance Objectives.* November 1996. (ETR 315.)

8 ETSI. *Speech Processing, Transmission and Quality Aspects (STQ); Overall Transmission Plan Aspects for Telephony in a Private Network.* November 1998. (ETSI Guide 201 050.)

9 ETSI. *Mandatory speech codec; AMR speech codec; Interface to I_u and U_u.* (ETSI 3G TS 26.102 v 3.1.0 (2000-03), Release 1999.)

10 De Vleeschauwer, D et al. Quality Bounds for Packetized Voice Transport. *Alcatel Telecom Review*, First quarter 2000, 19–23.

11 De Vleeschauwer, D et al. Determining the Number of Packet-based Phones that can be Supported by One Access Node. In: *Proceedings of the 14th ITC Specialists Seminar on Access Networks and Systems*, Girona (Spain), 25–27 April 2001, 197–204.

12 Goderis, D. *Functional Architecture Definition and Top Level Design.* 11 September 2000. (IST project Tequila deliverable D1.1.) [online] – URL: http://www.ist-tequila.org/deliverables/d11_final.pdf

13 Goderis, D et al. *Service Level Specification Semantics, Parameters and Negotiation Requirements.* November 2000. (IETF Internet Draft.) draft-tequila-diffserv-sls-00.txt

14 Poppe, F, De Vleeschauwer, D, Petit, G H. Guaranteeing Quality of Service to Packetised Voice over the UMTS air Interface. In: *Proceedings of the Eighth International Workshop on Quality-of-Service (IWQoS 2000),* Pittsburgh (PA), 5–7 June 2000.

15 Poppe, F, De Vleeschauwer, D, Petit, G H. Choosing the UMTS Air Interface Parameters, the Voice Packet Size and the Dejittering Delay for a Voice-over-IP Call between a UMTS and a PSTN Party. In: *Proceedings of IEEE Infocom 2001,* Anchorage (AK), 22–26 April 2001, 805–814.

16 Johannesson, N O. The ETSI Computation Model: A Tool for Transmission Planning of Telephone Networks. *IEEE Communications Magazine,* 35 (1), 70–79, 1997.

17 Van Hoey, G et al. Capacity planning strategies for voice-over-IP traffic in the core network. In: *Proceedings of the IEEE Workshop on High Performance Switching and Routing (HPSR2001),* Dallas (Texas), 29–31 May 2001, 109–113.

Quality of Service in UMTS

THOR GUNNAR ESKEDAL AND FRÉDÉRIC PAINT

The third generation cellular network, UMTS (Universal Mobile Telecommunication System), is a standard specified by the Third Generation Partnership Project (3GPP). It deploys the 3rd generation W-CDMA air interface which will be deployed in Europe and Asia, including Japan and Korea in the frequency band around 2 GHz. With this frequency it is capable of a peak bandwidth of 2 Mb/s which may support simultaneous low bit rate voice services and high bit rate multimedia and video applications. To ensure reusability of platforms, it was decided to reuse the GPRS architecture. With the advent of real time Internet multimedia services it was felt necessary to ensure the provisioning of QoS. This was not trivial given the deficiencies of GPRS in supporting QoS. Considerable work was made to provide the necessary enhancements in the specifications to ensure adequate QoS support.

In this paper we provide an overview of the UMTS Quality of Service concept and give some insight into the mechanisms used to provide Quality of Service to upper layers. Additionally we discuss the latest enhancements being included in UMTS specifications to provide end-to-end Quality of Service.

Thor Gunnar Eskedal (36) obtained his Masters degree in physics in 1990 from the University of Oslo. After working one year as a senior research assistant in the Department of Telematics at the Norwegian University of Science and Technology (NTNU), he joined the broadband networks group at Telenor R&D in 1991. In the last couple of years he has been involved in the UMTS standardisation effort mainly concentrating on IP-based QoS mechanisms, IP mobility and systems architecture.

thor-gunnar.eskedal @telenor.com

Frédéric Paint (28) has been working for Telenor R&D since his graduation from ENST Paris (telecom engineering school) in 1998. His work has focused on 3G core networks and their evolution. This effort included participation in research projects (e.g. Eurescom P920 and Eurescom P1013) and standardisation activities (3GPP). More recently he has been involved in the field of mobility in IP networks specifically on micro-mobility support and inter-access mobility.

frederic.paint@telenor.com

1 Introduction

The scope of this article is to give a brief overview of the QoS framework of the 3rd generation mobile network UMTS (Universal Mobile Telecommunication System). UMTS is developed by the 3GPP (Third Generation Partnership Project), which is an interest organisation put together of many global standardisation bodies such as ETSI, ARIB, T1, TTC, CWTS and others. The vision of UMTS is to support both traditional circuit switched and packet based services as well as multimedia. All services shall be carried over the W-CDMA (Wideband Code Division Multiple Access) radio system, which is specially suited to support variable bit rates and traffic with different characteristics. With the tremendous growth of the Internet the past few years, and the wide variety of new IP based applications that have appeared, it is recognised that the new mobile network also has to be equipped to support the application requirements. People would like to be able to access the same services from a mobile terminal as from their stationary terminal, with the same or only minor degradation in quality. Major effort has therefore been put on developing a future proof QoS framework capable of supporting a wide spectrum of applications, taking into account the special characteristics of radio transmission. Admission control with good resource management is vital to achieve acceptable quality for the users connected to the system. When users move between radio cells (handovers) functionality has to be deployed which quickly manages to switch the connecting point of the user to the network without noticeable degradation of quality. The radio interface puts major challenges on the development of a QoS control system to cope with the scarce resources, unpredictable traffic picture, vulnerability to air disturbances, etc. The work is therefore tedious with very many aspects to consider. Since the UMTS systems evolve through releases the QoS framework is updated accordingly.

The first release of UMTS was Release 99. That release was finalised in June 2000. Rel-4 was the next release, which was completed in March 2001. 3GPP is now working on Rel-5. Rel-5 is expected Q 1 of 2002. The UMTS architecture has developed through these releases, becoming gradually more IP focused both regarding transport and signalling. The circuit switched domain comprising much of the legacy GSM core network infrastructure has gradually been replace by an optionally IP based transport infrastructure. Hence, the major impact of IP based applications and the importance of real time support on an IP infrastructure have been recognised. A targeted goal has been to support VoIP and packet based multimedia applications on the packet switched domain with the same or better quality than experienced in the circuit switched GSM network.

The article starts by giving a brief overview of the different UMTS releases. The different releases impact the QoS developments by gradually extending the QoS functionality. The different releases cover the UMTS QoS bearers, QoS management entities, end-to-end aspects, QoS for multimedia handling, policy framework etc., making up a total QoS framework. Chapter 3 looks specifically into the QoS framework and explains the QoS architecture of UMTS both related to the control and user plane functionality. The mechanisms used in UMTS to support differentiated services with separate service

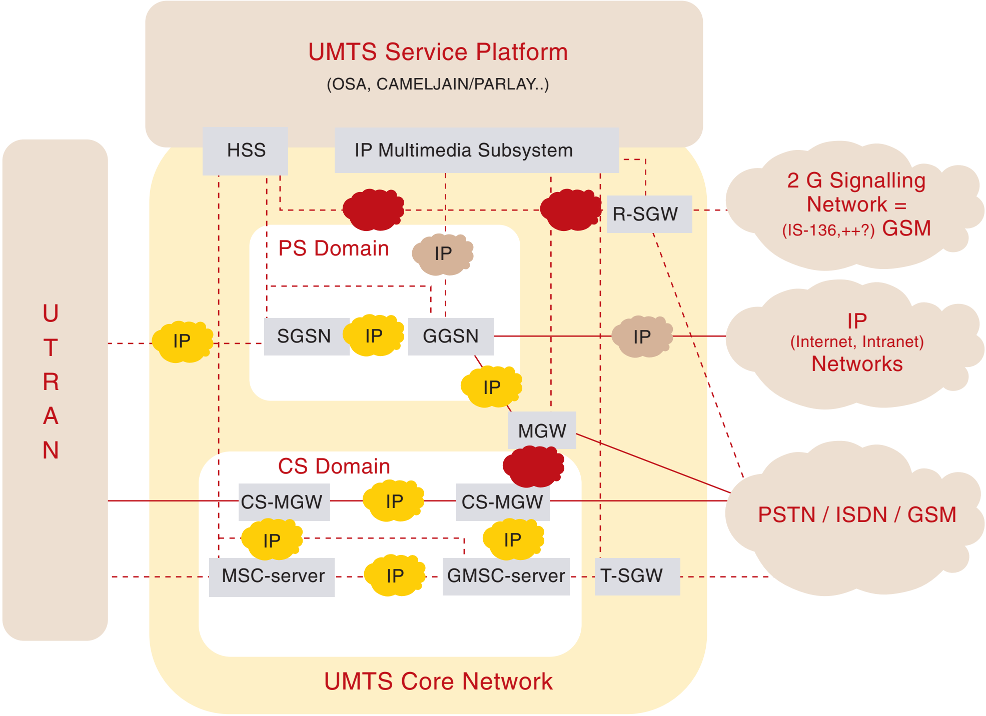


Figure 2-1 UMTS Architecture

classes will be described as well as aspects concerning QoS support over the radio interface, handover issues and interworking with legacy mobile systems. Chapter 4 gives a status overview of the end-to-end capabilities in UMTS including the policy framework and IP layer functionality to negotiate end-to-end QoS. To conclude we give a summary of the matters discussed and point out some remaining open standardisation issues.

2 UMTS Architecture Overview

UMTS has been standardised in phases. The initial release (R99) includes the basic functionality to access IP networks and the circuit switched networks. The Core network part is composed of a packet domain based on the GPRS architecture and a circuit domain based on the GSM architecture. The radio network (UTRAN) is based on the W-CDMA technology and is functionally independent from the core network. Its function is to provide access to the core network domains via the Iu interface. Further releases add features to this basic functionality set. Release 5 introduces the IP multimedia subsystem for efficiently providing IP based multimedia services over the packet domain. The IP Multimedia subsystem is an overlay control system to perform session control of IP multimedia services based on SIP signalling. The transport part is evolving towards an "All IP" architecture, e.g. the transport for the CS domain could be transported by IP through the CN.

In the following we provide a short description of the core network, the radio network and the IP multimedia subsystem. Our focus will be on the PD of UMTS.

2.1 Core Network

The Core Network [5,6,7] transports packets between the radio access network and external IP networks. Besides this transport function it is also responsible for Lawful Intercept, Charging, authenticating users and authorising connections to the external IP networks as well as functionality specific to mobile networks such as mobility management. The three main nodes of the Core Network are the Home Subscriber Server (HSS), the Serving GPRS support node (SGSN) and the Gateway GPRS Support Node (GGSN).

The Home Subscriber Server (HSS) is the master database for a given user. It is the entity containing the subscription related information to support the network entities actually handling calls/sessions. The HSS includes the GSM HLR functionality and IETF AAA functionality.

The SGSN is the node that serves the MS. The SGSN supports GPRS for GSM and/or UMTS. When the mobile is attaching to the network, the SGSN establishes a mobility management context that records the mobility and security information of the mobile. When the mobile wants to establish a connection to external networks it initiates PDP Context Activation procedure. The SGSN participates in that procedure and keeps track of the parameters (Routing information, QoS information) of that connection in a PDP context.

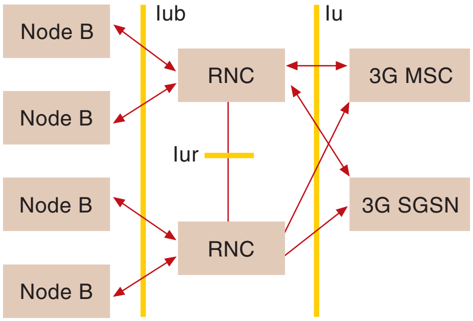


Figure 2-2 The UTRAN reference architecture

The GGSN is the node connecting the UMTS network to external IP networks. It contains routing information to reach attached users and interfaces various control nodes, e.g. to support authentication and find the location of mobile users. The routing information is used to tunnel data packets to the MS's current point of attachment, i.e. the SGSN. The GGSN is also responsible for allocating IP addresses to the terminals.

The SGSNs and GGSNs of an operator are connected together by an IP network. Packets are tunnelled between those nodes using the GPRS tunnelling protocol (GTP) which runs over the IP protocol layer. DiffServ is used to provide QoS differentiation within this IP network. The HSS is connected to the SGSNs by an SS7 network.

2.2 UTRAN

The access network in UMTS, the UTRAN (UMTS Terrestrial Radio Access Network) [8–21], is the new revolutionary part from GSM/2+ supporting a data-rate of up to 2 Mb/s. This data rate is achieved in UMTS by deploying W-CDMA (Wideband-Code Division Multiple Access) technology deploying the 2.2 – 2.4 GHz frequency band. With this new radio technology all services share a common radio resource deploying spread spectrum technology with a "usage on demand" approach. This is spectrum efficient but challenging regarding support of QoS. Since it is a common resource the bandwidth for each user may vary. Apart from the number of simultaneous users the data rate is dependent on the users' distance from a base transceiver, the velocity of the users as well as different air disturbances as random noise, weather condition, etc.

The UTRAN (Figure 2-2) is comprised of two distinct nodes, the Radio Network Controller (RNC) and the Node B. The Node B is the radio transceiver station in UMTS. Each node B may have several radio transceivers and each RNC may control up to 64 Node Bs. The RNC is in charge of the radio resources and allocates radio channels to different types of traffic according to their demand of resources and service requirements.

The physical transport varies across the different interfaces. Across the Iub interface (ref. Figure 2-2) AAL2/ATM is deployed to transport radio frames between Node B and RNC. No distinction is made between data or real time traffic across the Iub interface. Since there is no differentiation of traffic types across the Iub interface only one ATM QoS class may be deployed. At the RNC the radio frames are reassembled and transported to the SGSN. In handover cases where a mobile moves out of reach from the serving RNC over to another RNC's control domain traffic is routed across the Iur interface. This interface therefore ties two RNCs together and thereby avoids including the Core network in these handover cases. This optimises the QoS since only a redirection handling is needed to maintain the connection. The "old" RNC functions as the anchor RNC and still controls the session/call. Also across the Iur interface AAL2/ ATM is used for transport of both data and real time traffic. Across the Iu interface, i.e. the interface between the RNC and the MSC and the SGSN, data traffic is transported on IP over AAL5/ATM. Circuit switched traffic is transported directly on AAL2/ATM to/from the 3G-MSC.

As seen, the UTRAN heavily deploys ATM as the underlying transport both for packet data and circuit switched data. Several arguments have been put forward regarding the use of ATM in the RAN. People claiming that the bandwidth in the RAN would be scarce argue that ATM is the most flexible and best suited technology to support service differentiation and optimal usage of bandwidth (at least on the Iub). They also argue that by specifying ATM under IP, it puts the operators in a position to choose to use ATM QoS mechanisms or IP based QoS. Today ATM is used in access networks, but it is foreseen that with IP based QoS mechanisms ATM may be obsolete in a couple of years. Much work is therefore conducted in 3GPP to look into different possibilities for usage of IP based protocols in the RAN to try to substitute the ATM protocol stacks and still support the QoS requirements.

2.3 The IP Multimedia Subsystem

The IP Multimedia Subsystem [3] gives better support for value added services such as multimedia, multimedia messaging, global text telephony, push services, etc. Many new functional entities compose this system. Most of them are related to call control and service control for multimedia sessions.

The main entity is the Call State Control Function (CSCF). The CSCF is responsible for the call control part of the IP multimedia services. It is very similar to a SIP server deploying the Session Initiation Protocol (SIP).

To support the IP multimedia traffic SIP is chosen to carry the call control signalling peer-to-peer. This protocol family is looked upon as the most promising multimedia protocol as the trend has turned toward a major growth in IP based applications and multimedia. SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions. Conceptually it inherits features from other IETF protocols, in particular Simple Mail Transfer Protocol (SMTP) and Hyper Text Transfer Protocol (HTTP). The SIP is a textual protocol based on a client-server model. Both the client and server parts of SIP are however implemented in a user terminal.

The Multimedia Resource Function (MRF) performs multiparty call and multimedia conferencing functions. MRF would have similar functions as an MCU in an H.323 network.

It is also responsible for bearer control (with GGSN) in case of multiparty/multimedia conferences. It may also communicate with the CSCF for service validation for multiparty/multimedia sessions. Other entities are used to provide interworking with legacy circuit switched networks (GSM/PSTN/ISDN).

The next chapter presents the QoS framework of UMTS [1,2]. The development of this framework has been done incrementally. In the first release of UMTS only internal UMTS QoS is provided, i.e. QoS from the mobile termination to the Gateway (GGSN). The next releases develop the framework further by including end-to-end QoS support.

3 UMTS QoS Architecture

3.1 UMTS QoS development

The UMTS QoS framework is being established based on a set of requirements of both general and technical character. The requirements incorporate internal as well as the end-to-end aspects. Some of these requirements are given in Figure 3-1.

During the work these QoS requirements act as strict working rules for the QoS team in 3GPP. However, before starting the QoS specification, a system perspective had to be clarified in terms of which bearer services the UMTS system contained, and how they interacted. The bearers comprise a framework of how the QoS functional entities interact. It also showed how the bearers supported each other regarding what information had to be conveyed both vertically and horizontally to establish a QoS path across the UMTS network as well as end-to-end.

To ensure QoS across the UMTS network, various attributes, e.g. error tolerance, delay values, SDU sizes, were described as guidelines for the real implementation. It was crucial that the UMTS system was seen as a network with internal budgets on e.g. delay and jitter and that the end-to-end QoS requirements between two communicating subscribers would be met. Delay budgets for the different network segments were derived from several studies on customer satisfaction as well as the feasibility of the network components and transmission links. To differentiate between different traffic requirements, specific UMTS QoS classes were described. These classes should ensure that the characteristics and requirements of each individual traffic flow would be met.

The UMTS system should interact with legacy networks. This interoperability between different wireless systems was felt necessary given that many operators have large investments in 2G systems. QoS parameters therefore have to be mapped between the different systems, and map-

Figure 3-1 Technical requirements regarding QoS for UMTS

Requirements for UMTS QoS

- UMTS shall provide QoS attribute control on a peer-to-peer basis between UE and 3G gateway node;
- UMS QoS shall provide a mapping between applications requirements and UMTS services;
- UMTS QoS shall be able to efficiently interwork with current QoS schemes. Further, the QoS concept should be capable of providing different levels of QoS by using UMTS specific control mechanisms (not related to QoS mechanisms in the external networks);
- A session based approach needs to be adopted for all packet mode communication within the 3G serving node with which UMTS QoS approach shall be intimately linked, essential features are multiple QoS streams per address;
- The overhead and additional complexity caused by the QoS scheme should be kept reasonably low, as well as the amount of state information transmitted and stored in the network;
- QoS shall support efficient resource utilisation;
- The QoS attributes are needed to support asymmetric bearers;
- Applications (or special software in UE or 3G gateway node) should be able to indicate QoS values for their data transmissions;
- QoS behaviour should be dynamic, i.e. it shall be possible to modify QoS attributes during an active session;
- Number of attributes should be kept reasonably low (increasing number of attributes, increased system complexity);
- User QoS requirements shall be satisfied by the system, including when change of SGSN within the Core Network occurs.

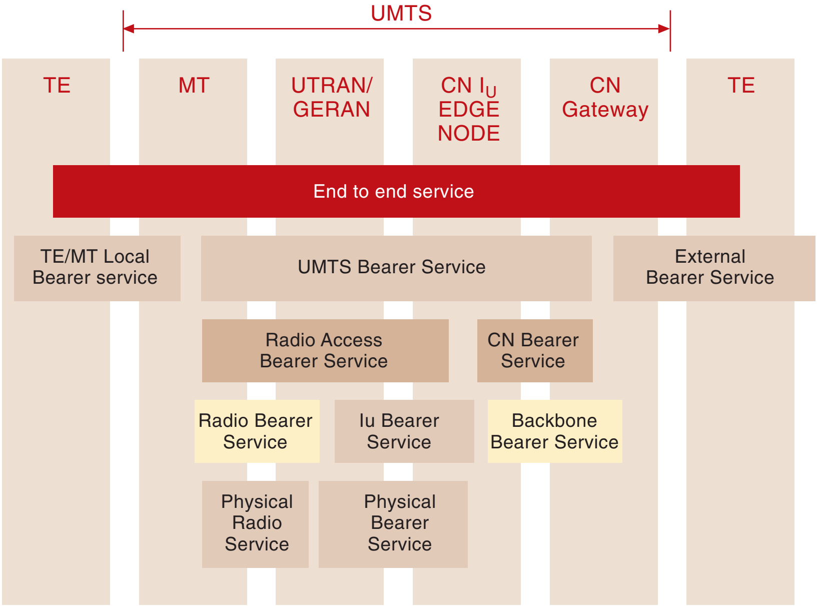


Figure 3-2 UMTS QoS bearers

ping schemes were worked out between UMTS and GPRS, and between UMTS and GSM.

As the UMTS internal QoS mechanisms became stable the end-to-end aspects came more into focus. A policy framework was developed to support policy rules for the end-to-end communication on the IP layer. A goal was to make the interworking with the external networks as seamless and efficient as possible with high flexibility to convey end-to-end QoS information. To support QoS related to the IP multimedia subsystem, SDP (Session Description Protocol) was used to carry the session parameters. SDP is a protocol with the SIP family which described the resource requirements for the session.

When discussing UMTS QoS mechanisms it is important to bear in mind the strong relationship between UMTS packet domain QoS and IETF based QoS mechanisms. It was a goal for 3GPP to conform as much as possible to IETF based QoS mechanisms to avoid duplicating work, and try to provide a smooth and efficient integration/ interworking with external IP based networks. UMTS therefore adopted DiffServ at the connectivity IP level between RNC and SGSN and between SGSN and GGSN to support differentiated services. Interworking between UMTS specific QoS mechanisms and IETF based QoS mechanisms such as RSVP and DiffServ therefore became an important work item. Work is still on-going in 3GPP around these issues.

In the following we will go into more detail on the internal QoS provisioning in UMTS. End-to-end issues are addressed in Chapter 4.

3.2 UMTS Bearers

UMTS specifies different levels of QoS. These levels are specified with different bearer services as depicted in Figure 3-2. As shown, the UMTS system comprises the nodes from the MT to the CN Gateway. Within this network the QoS will be ensured by means of different bearer services. Each bearer service will deploy the services supported by the layer below. The QoS across the UMTS network is specified by the UMTS bearer services comprising the QoS handling in the UTRAN and in the Core network. The UMTS bearer services interact with the local bearer services on the terminal side and the external bearer services on the border towards external networks. Together these bearer services deliver the end-to-end service comprising the end-to-end QoS that the user experiences.

The QoS across the UTRAN is supported by the RAB (Radio Access Bearer). The RAB is supported by the Iu bearer service and the Radio bearer service. The W-CDMA based Radio bearer comprises the physical wireless transport. The RAB service may differentiate between services classed by deploying different radio channels, e.g. dedicated channels or shared channels. Management of the channels regarding e.g. error control, priority of radio resources etc. varies between the channel types. Different channel types may therefore be utilised for different traffic demanding different QoS requirements. Real time traffic most often deploys dedicated channels. The Iu bearer service together with the physical bearer service comprise the services across the Iub, Iur and Iu interface. An ATM protocol stack is used across the Iub and Iur interface as explained in Chapter 2. Across the Iu interface it is optional to use ATM QoS mechanisms and/or IP based QoS mechanisms. If one deploys IP QoS, DiffServ shall be the mechanisms to support QoS across the Iu interface.

The Core network bearer service for the packet domain is pretty similar to the Iu bearer service. At the IP connectivity layer DiffServ handles service differentiation and QoS support. The layers beneath the IP level is not specified for the CN part as it is for the Iu interface.

3.3 QoS Management

To support the required QoS throughout the UMTS network, and to be able to allocate and keep track of resource usage internally, each UMTS entity has to perform certain QoS related management functions. UMTS specifies management functions for both the control and user plane. Figure 2 depicts the QoS management functions for the control plane.

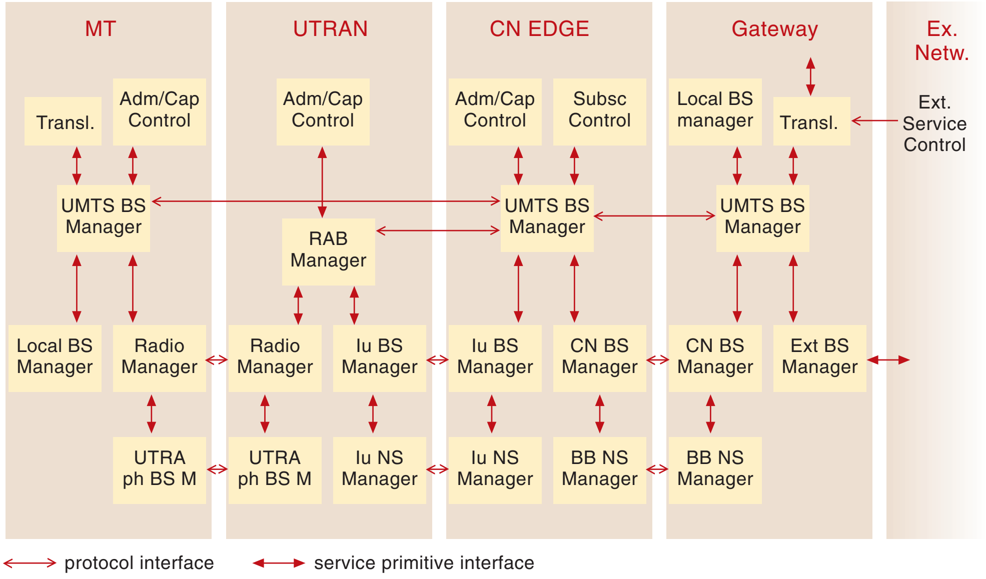


Figure 3-3 QoS management functions for the UMTS bearer service in the control plane

3.3.1 QoS Management for the Control Plane

The *UMTS BS Manager* in the UE, CN EDGE and the Gateway signal between each other and via the translation function with external instances to establish or modify a UMTS bearer service. Each of the UMTS BS managers interrogates its associated admission/capability control whether the network entity supports the requested service and whether the required resources are available. Additionally, the CN EDGE UMTS BS manager verifies with the Subscription Control the administrative rights for using the service.

In requesting a service the UMTS BS manager of the CN EDGE translates the UMTS bearer service attributes into Radio Access Bearer (RAB) service attributes, Iu bearer service attributes and CN bearer service attributes.

The *RAB manager* verifies with its admission/capability control whether the UTRAN supports the specific requested service and whether the required resources are available. It translates the RAB service attributes into radio bearer service and Iu bearer service attributes and requests the *radio BS manager* and the *Iu BS manager* to provide bearer services with the required attributes.

The Gateway UMTS BS manager translates the UMTS bearer service attributes into CN bearer service attributes and requests its *CN BS manager* to provide the service. To support the transport into external environments, the UMTS BS manager communicates with a translation function who translates the UMTS bearer service attributes into the external bearer services and requests the service from the *Ext BS manager*.

Radio BS managers, Iu BS managers and CN BS managers use services provided by lower layers as indicated in Figure 3-2.

Admission/Capability control maintains information about all available resources of a network entity and about all resources allocated to UMTS bearer services. It determines for each UMTS bearer service request or modification whether the required resources can be provided by this entity, and it reserves these resources if allocated to the UMTS bearer service. The function also checks the capabilities of the network entity to provide the requested service, i.e. whether the specific service is implemented and not blocked e.g. for administrative reasons.

3.3.2 QoS Management for the User Plane

Figure 3-4 depicts the functions taking place prior to or during user data transport through the UMTS network. The user traffic traverses several UMTS specific functions to adapt the traffic to the UMTS transport functionality. The functions can be categorised into the following entities:

- *Classification:* The classification function (Class), which is located both in the Gateway and in the UE, assigns user data units received from the external bearer service or internal service interface to the appropriate UMTS bearer service. This is done according to the QoS requirements of each user data unit.

- *Conditioning:* The traffic conditioner (Cond.) in the UE provides conformance of the uplink user data traffic with the QoS attributes of the relevant UMTS bearer service. In the Gateway a traffic conditioner may provide conformance

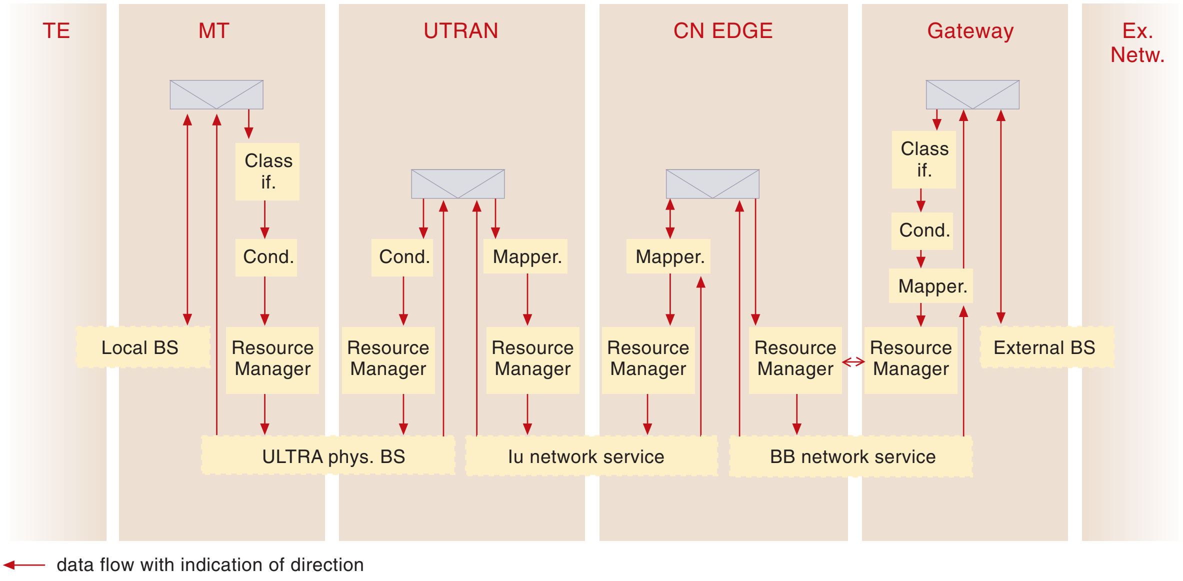


Figure 3-4 QoS management functions for the UMTS bearer service in the User plane

of the downlink user data traffic with the QoS attributes of the relevant UMTS bearer service. For example, the packet-oriented transport of the downlink data units is received from the external bearer service and sent through the core network to the UTRAN and is buffered at the RNC. If the downlink traffic results in bursts of data units not conformant with the UMTS BS QoS attributes, a traffic conditioner in the UTRAN conforms the data traffic according to the relevant QoS attributes as e.g. the peak bandwidth limit.

The traffic conditioner is not necessarily the only function to ensure that the traffic does not exceed the QoS attributes. For example a resource manager may also provide conformance with the relevant QoS attributes by appropriate data unit scheduling. Or, if fixed resources are dedicated to one bearer service the resource limitations implicitly condition the traffic.

- *Mapping:* The mapping function (Mapper) marks each data unit with the specific QoS indication related to the bearer service performing the transfer of the data unit. This may be marking the packets with specific DiffServ code points for differentiated treatment in DiffServ enabled IP networks.

- *Resource manager:* Each of the Resource Managers of a network entity is responsible for a specific resource. The resource manager distributes its resource budget between all bearer services requesting transfer of data units. The resource manager thereby attempts to support the QoS attributes required for each individual bearer service.

3.4 UMTS QoS Classes

The network must be able to distinguish between types of services to be able to support different QoS requirements. Four QoS classes have been specified:

- *Conversational class*
  The conversational class supports real-time communication between entities. The class provides low latency and drop reliability.

- *Streaming class*
  The streaming class intends to support applications which are not real time demanding but sensitive to jitter. However, the latency between the communication entities must be limited within a defined maximum value.

- *Interactive class*
  The interactive class offers three levels of precedence and supports non real-time applications.

- *Background*
  The background class supports non real-time demands. The class is served with the lowest priorities.

As noted the difference between them is first and foremost their delay sensitivity, ref. Table 1. The conversational class is the most delay sensitive. This class is therefore best suited for real time services such as voice applications. The background class is the most delay intolerant class and is suited for e-mails and file transfer. The table on the following page gives an overview of the classes and what types of services are suitable to use within each class.

Table 1 UMTS QoS classes and their characteristics

Traffic class	Conversational class Conversational Real Time (RT)	Streaming class streaming RT	Interactive class Interactive best effort	Background class Background best effort
Fundamental characteristics	Preserve time relation (variation) between information entities of the stream - Conversational - pattern (stringent - and low delay)	Preserve time relation (variation) between information entities of the stream	Request response pattern Preserve payload content	Destination is not expecting the data within a certain time Preserve payload content
Example of the application	voice	streaming video	web browsing	Background download of e-mails

Each of the QoS classes in Table 1 are associated with a set of QoS parameters describing the requirements demanded from the associated applications. Table 2 describes the QoS attributes and Table 3 gives the value ranges of UMTS bearer service attributes for the different QoS classes. These values give the capabilities demanded from the UMTS network. Value ranges for the Radio access bearer have also been put forward. Some parameters, e.g. transfer delay, will contain a lower maximum value for the Radio access due to the delay budget through the Core network. For asymmetric bearers some attributes may have different values in uplink than in downlink direction.

3.5 UMTS Bearer Establishment

To illustrate how the different parts work together we present the overall procedure for establishing a UMTS bearer. We also discuss how QoS is mapped to DiffServ within the IP transport networks. Additionally we address the

Traffic class	UMTS QoS classes like conversational, streaming.
Maximum bitrate (kb/s)	Maximum no. of bits provided by the UMTS network within a period of time.
Guaranteed bitrate	Guaranteed number of bits delivered by the UMTS network within a period of time. A value of k greater than one Maximum SDU (Service Data Unit) size may be specified in releases beyond R99 to capture burstiness of sources.
Deliver order (y/n)	In-sequence delivery of SDU packets or not.
Maximum SDU size (octets)	The maximum size of SDUs.
SDU format information (bits)	List of possible exact sizes of SDU.
SDU error ratio	Fraction of SDUs lost or deteced as erroneous.
Residual bit error ratio	Undected bit error ratio in the delivered SDUs.
Delivery of erroneous SDUs (y/n)	Whether SDUs detected as erroneous shall be delivered or discarded or not.
Transfer delay (ms)	Maximum delay for 95th percentile of the distribution of delay for all delivered SDUs. Delay is time from a request to transfer an SDU at one SAP to its delivery at the other SA.
Traffic handling priority	Relative importance for handling of all SDUs belonging to the UMTS bearer compared to the SDUs of other bearers.
Allocation/retention priority	Used for differentiating between bearers when performing allocation and retention of a bearer.

Table 2 UMTS Bearer Service attributes

Traffic class	Conversational class	Streaming class	Interactive class	Background class
Maximum bitrate (kb/s)	< 2 048	< 2 048	< 2 048 - overhead	< 2 048 - overhead
Delivery order	Yes/No	Yes/No	Yes/No	Yes/No
Maximum SDU size (octets)	<= 1 500 or 1 502	<= 1 500 or 1 502	<= 1 500 or 1 502	<= 1 500 or 1 502
SDU format information				
Delivery of erroneous SDUs	Yes/No/ -	Yes/No/ -	Yes/No/ -	Yes/No/ -
SDU Residual BER	$5*10^{-2}$, 10^{-2}, $5*10^{-3}$, 10^{-3}, 10^{-4}, 10^{-6}	$5*10^{-2}$, 10^{-2}, $5*10^{-3}$, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}	$4*10^{-3}$, 10^{-5}, $6*10^{-8}$	$4*10^{-3}$, 10^{-5}, $6*10^{-8}$
SDU error ratio	10^{-2}, $7*10^{-3}$, 10^{-3}, 10^{-4}, 10^{-5}	10^{-1}, 10^{-2}, $7*10^{-3}$, 10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-6}	10^{-3}, 10^{-4}, 10^{-6}
Transfer delay (ms)	100 -> maximum value FFS	250 -> maximum value		
Guaranteed bit rate (kb/s)	< 2 048	< 2 048		
Traffic handling priority			1, 2, 3	
Allocation/Retention priority	1, 2, 3	1, 2, 3	1, 2, 3	1, 2, 3

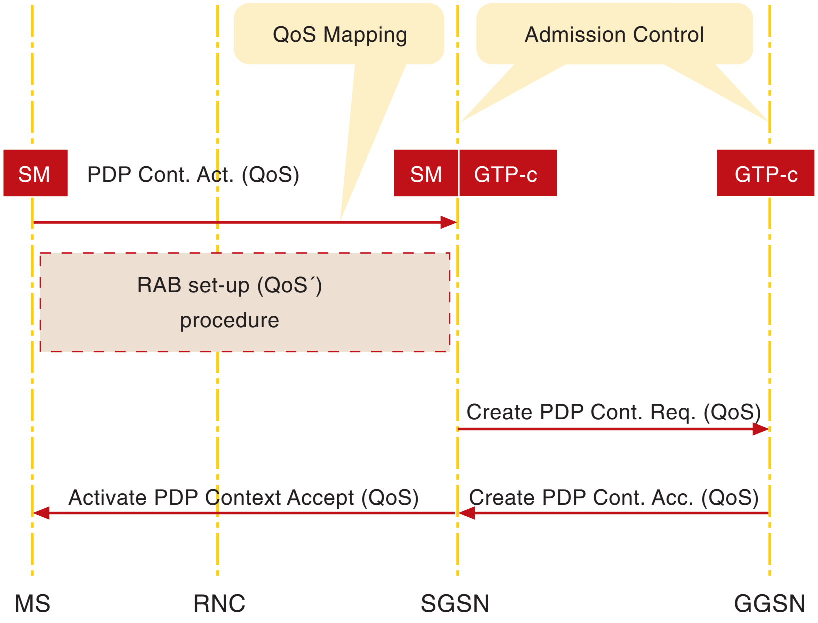
Table 3 Values for the UMTS bearer service attributes

establishment of the radio access bearer in more detail.

3.5.1 Overall Procedure

Figure 3-5 shows how the bearer is established in UMTS. The user establishes a UMTS bearer (PDP context activation) by specifying the QoS requested. The SGSN authorises the QoS given the subscription of the user and does admission control given its available resources. It then requests the establishment of a radio access bearer (RAB). To do so it derives the adequate QoS profile for the RAB given the QoS profile of the UMTS bearer. As an example the transfer delay might be set to 80 ms if the UMTS bearer QoS profile indicates 100 ms. This mapping is implementation dependent.

Figure 3-5 UMTS Bearer establishment



Once the Radio Access Bearer has been set-up the SGSN forwards the request to the GGSN. The GGSN will perform admission control and replies to the SGSN that in turn replies to the Terminal. Additionally, the SGSN will take the initiative to configure the Core network bearer. The QoS proposed by the network may be lower than that requested by the terminal. The terminal might reject the establishment of the UMTS bearer, renegotiate the QoS or accept the QoS.

3.5.2 QoS across IP Transport Networks

The UMTS architecture for the PS domain deploys IP transport networks for the Iu interface (RNC – SGSN) and for the Gn interface (SGSN-GGSN). Across these interfaces DiffServ is used as the IP transport layer QoS mechanism. Hence, at the RNC, SGSN and GGSN the QoS mechanisms at the UMTS bearer service layer are mapped to DiffServ Code Points to preserve the QoS relationship between the protocol layers. An important task is to configure the mapping the best possible way. Since QoS parameter mapping and configuration of the QoS mechanisms may depend on different issues, e.g. operator strategy, these mappings are not specified by 3GPP. Also the scheduling mechanism in the UMTS nodes has not been specified by 3GPP and will be vendor specific. It has been recognised that it is important for each operator to be able to configure the QoS parameters with regard to its own policies of customer behaviour. This includes how the system will handle situation such as heavy traffic, congestion, faults, as well as the actions against misbehaving users and the impact on pricing rates.

An example of mapping between UMTS classes and DiffServ code points is given in Table 4.

UMTS QoS class	DiffServ class
Conversational	Expedited forwarding
Streaming	Assured forwarding (AF11, AF12, AF13)
Interactive priority 1	Assured forwarding (AF21, AF22, AF23)
Interactive priority 2	Assured forwarding (AF31, AF32, AF33)
Interactive priority 3	Assured forwarding (AF41, AF42, AF43)
Background	Best effort

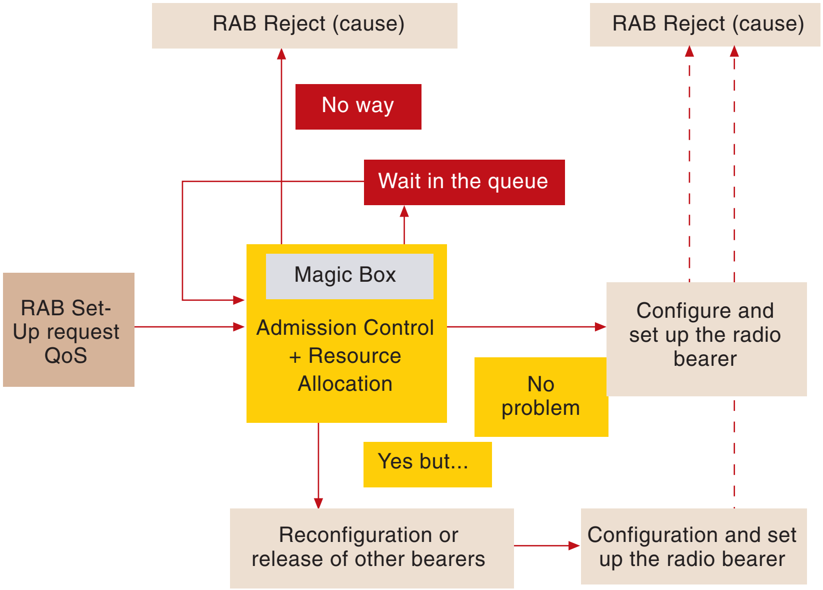
Table 4 QoS class mapping between UMTS QoS and DiffServ Code point

3.5.3 QoS across the Radio Interface

The Radio Network Controller (RNC) is responsible for the allocation, management and termination of radio bearers. Radio bearers are established when a radio access bearer establishment is requested. The RNC first determines whether there are enough resources to service the request and if not, it may degrade an existing radio access bearer with lower priority so as to allow the newcomer. Alternatively, it will reject the request. When the resources are available the resource manager selects the appropriate radio bearer to establish according to the values of the parameters specified in the RAB establishment request. A radio bearer is characterised by the type of channel it is using, the parameters describing this channel and the configuration of the radio protocols.

There are two main types of channels, dedicated channels for time stringent traffic and shared channels for non time stringent traffic. For a dedicated channel the access to this channel is restricted to the owner of the bearer. The channel is also characterised by the frequency and the CDMA codes. The code defines the raw data-rate on the channel. Error coding is always used and additional redundancy is provided at the radio link layer control by a retransmission protocol. The choice of the error coding code and whether to use retransmissions or not depend on the level of reliability needed for the radio bearer and the delay requirements. The mapping algorithm of the QoS given in the radio access bearer

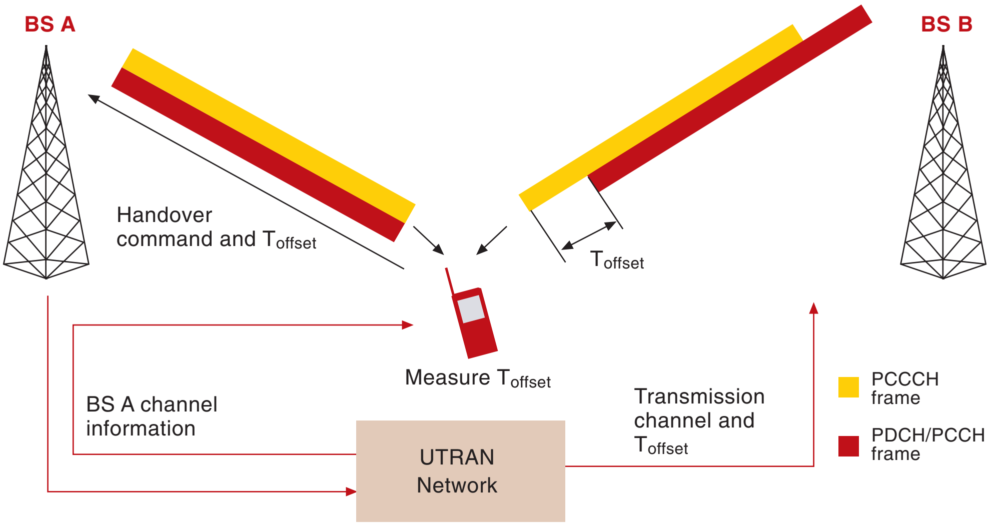
Figure 3-6 Radio Bearer establishment


RAB Reject (cause)
RAB Reject (cause)
No way
Wait in the queue
Magic Box
RAB Set-Up request QoS
Admission Control + Resource Allocation
Configure and set up the radio bearer
No problem
Yes but...
Reconfiguration or release of other bearers
Configuration and set up the radio bearer

QoS	Transport Channel	Type of service	Radio protocols
Conversational 8 kb/s guaranteed bit rate	Dedicated uplink and downlink, 8 kb/s	Low delay, high priority, low reliability	No retransmissions (RLC transparent)
Streaming 64 kb/s video	Dedicated downlink 64 kb/s and packet channel uplink	Guaranteed delay, high priority, low reliability	No retransmissions (RLC transparent)
Interactive mean bit rate 64 kb/s on downlink	Shared channel uplink 16 kb/s and downlink 144 kb/s	Mean data rate guaranteed, high reliability	Retransmissions

Table 5 Mapping of QoS to Radio Bearer

Figure 3-7 Soft handover in UMTS



set-up to a specific radio bearer is not part of the standards and is left for the implementation and deployment. We illustrate in Table 5 one simple algorithm taken from [22]. The Iu bearer is set-up after the radio bearer has been established.

3.6 QoS and Handovers

3.6.1 UMTS Handovers

Most wireless systems need mechanisms to support situations where radio connections allocated to mobiles moved out of reach of the radio transceiver they initiated the communication session from. To be able to keep the session alive it has to be taken over by another radio transceiver. This switching of radio transceiver is denoted handover. In UMTS there are two fundamentally different ways of conducting handovers – hard handover and soft handover. With hard handover the communication session is broken when a mobile leaves a radio transceiver and established again at the new radio transceiver. This switching of radio transceiver introduces a slight drop of connection and possible drop of packets. The user may hear it as a clip in a speech conversation. In UMTS this handover mode is comparable with that of GSM today regarding QoS. The radio system of UMTS however is especially constructed to perform soft handover with macro diversity. With soft handover the mobile can send or receive on up to several radio channels simultaneously thereby increasing the QoS performance, however at the cost of used bandwidth. Figure 3-7 shows a mobile receiving data from two Node Bs, connected to the same RNC, simultaneously.

This handover mechanism enhances the QoS, especially for voice and real-time services, since the session will always be on. The RNC receives the data from both Node Bs and based on different QoS criteria chooses the best one to forward to the remote host. This mechanism is denoted macro diversity. Macro diversity may be used in both uplink and downlink. In uplink it is the RNC that chooses the best traffic stream, in downlink it is the mobile itself that selects the best traffic stream. Figure 3-7 illustrates a soft handover case.

The handover procedures are transparent to the Core network except in the case of a terminal moving from one RNC to another. In that case the Core network may have to re-establish the bearers to the new RNC. This procedure called SRNC relocation first establishes bearers on the new path before the switching is accomplished. It uses the make before break concept allowing faster switching of routes and thus less disruption.

3.6.2 Handover to Legacy Systems

UMTS was designed to allow smooth deployment. A typical scenario for the first years of UMTS deployment is that UMTS covers a limited number of cities while GPRS/GSM is nation wide. In that context users would like to seamlessly roam between the two accesses. Handover between UMTS and GPRS/GSM is thus an important feature.

To allow easy interworking the provisioning of QoS in GPRS was aligned with UMTS for the release 99 of GPRS. GPRS supports only two QoS classes, the Background class and the Interactive class. In case of handover between UMTS and GPRS QoS renegotiation may occur. This allows the user/application to decide whether it can accept lower Quality of Service. If it cannot the session will be broken.

4 End-to-end QoS Support in UMTS

In Chapter 3 we described the QoS functionality internally to the UMTS network and the different components necessary to support QoS for different traffic types. To be able to support end-to-end QoS there has to be mechanisms to translate the UMTS QoS parameters to external QoS mechanisms and protect the UMTS network from external networks, e.g. given by policy rules. These rules contain information regarding what traffic is allowed to enter the other operators' domains, how to ensure that the rules are not violated and what action to take if the rules are violated. As Figure 3-2 depicts, the UMTS bearer service interfaces the external bearer service at the GW node, i.e. the GGSN. The GGSN therefore acts as the EDGE node towards external networks. The GGSN has therefore been equipped with functionality to police traffic entering and leaving the UMTS network and (re)negotiate external resources by communicating with external resource managers, e.g. by means of RSVP signalling.

In the following we describe the main functionality for end-to-end QoS provisioning in UMTS. In particular, we describe the concept of the IP bearer manager and the IP policy framework applied to UMTS. Finally we address the problem of co-ordination of the call control with the bearer control.

4.1 IP Bearer Manager

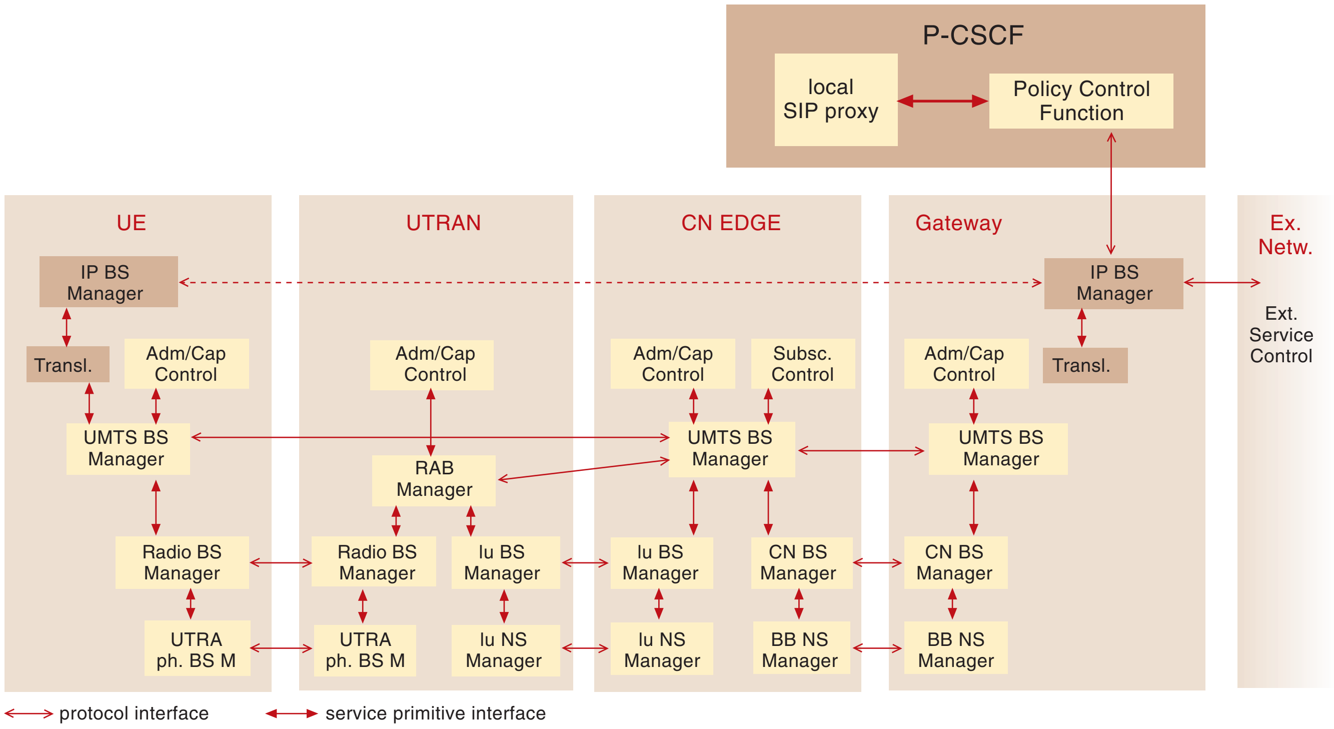
Figure 4-1 depicts the enhanced QoS framework indicating the new entities necessary for end-to-end IP layer QoS support. Both at the UE and at the GGSN a new functional entity known as the IP bearer manager is depicted. Not all UEs are expected to have implemented an IP bearer manager. For these terminals end-to-end QoS is provisioned by UMTS internal QoS mechanisms within the UMTS network (i.e. PDP context) and these are mapped to external QoS mechanisms at the GGSN. The IP BS Manager in the GGSN is used to control the external IP bearer service and communicate with the eventual UE's IP Bearer manager entity for end-to-end QoS (re)negotiation. As described earlier the UMTS bearer manager controls the QoS internally in the UMTS network. To support end-to-end QoS the UMTS bearer service manager has to communicate with the IP bearer service manager, and vice versa.

Due to the usage of different QoS mechanisms within the IP network, the IP bearer manager communicates with the UMTS BS manager through a translation function (ref Figure 4-1). The translation function translates the external QoS classification to UMTS specific QoS classification and vice versa. For example the UMTS QoS classes have to be translated into external QoS mechanisms, e.g. DiffServ code points. This translation is similar to the mapping function from UMTS QoS classes to/from the DiffServ implementation across the Iu and Gn interface as described in 3.5.2.

4.2 IP Layer UMTS Policy Framework

The policy framework in UMTS is constructed very similar to the IETF policy framework as described in IETF RFC 2753 "A Framework for Policy-Based Admission Control". The two main

Figure 4-1 QoS management functions for the end to end IP QoS



entities are the PCF (Policy Control Function) and the PEP (Policy Enforcement Point) which is a functionality of the IP BS manager in the GGSN. The COPS (Common Open Policy Service) protocol is used as the query and response protocol between the PCF and the PEP. The PCF is a logical policy decision element that uses standard IP mechanisms to implement policy in the IP bearer layer. Hence, it makes decisions with regard to network based IP policy using policy rules and communicates them to the IP bearer manager in the GGSN. Figure 4-1 depicts the end-to-end QoS management architecture.

The role of the PCF in regard to sessions is first and foremost to authorise the use of QoS resources to support a service to a specific user. The PCF may collect the parameters needed in several ways, e.g. by use of SIP signalling, i.e. from the SDP information, or the information in the RSVP FlowSpec. By use of the QoS information proper authorisation in the form of IP resources is communicated to the PEP. The PEP deploys this information to enforce the use of network recourses. If the user violates the resource usage, e.g. sends more packets than it is authorised to, it may drop the exceeding packets. If the user wishes to renegotiate the resource usage, new policy information has to be conveyed between the PCF and the PEP.

4.3 Bearer Control and Call Control Integration

This section gives a short overview of how bearer control and call control can be managed for a typical VoIP call. It is to be noted that further work is needed on these issues.

Voice over IP services over UMTS represent a major asset for a service provider. The provisioning of these services is nevertheless technically challenging in many regards. One of the issues that is being addressed in 3GPP is the co-ordination of the SIP call control procedure with the IP bearer establishment procedure. These two procedures need indeed to be co-ordinated to prevent

- Calls to be established prior to resources being reserved, thus leading to unnecessary call defects;
- Resources being committed before the call is set-up leading to inefficient use of resources and users being charged before the called party picks up;
- Theft of service and fraud.

A proposal to co-ordinate the call control with the bearer control is the two phase commit concept which has been designed in the context of cable networks. (Distributed Call Signalling (DCS) specification [24] and Dynamic Quality of Service specification (DQoS) [25].) The signalling procedure can be decomposed into four phases.

- Phase 1 establishes the call set-up state at both ends and authorises the traffic flow between both ends.
- Phase 2 reserves the resources (already authorised in phase 1) for that traffic flow.
- Phase 3 reports that the preconditions are met and ringing is accomplished.
- Phase 4 starts when the called party picks up the phone, the resources are committed.

The resource management framework distinguishes between two phases: Reserve and Commit. Reserving resources is the ability to admit the flow with the QoS requested. The commitment is the effective allocation of specific resources to the flow. Making this distinction improves system efficiency because it assigns resources only when necessary, and when the use of these resources may be charged to a customer.

UMTS does not support the distinction between resource reservation and resource commitment as the bearer set-up procedure includes both the admission of the bearer and the allocation of resources to that bearer. Some modifications of UMTS bearer establishment procedures are thus needed. The main concern is the radio bearer as the radio access network holds the scarce and expensive resources. One way of solving the problem is to design a radio access bearer reservation procedure. This would trigger the admission control in the RNC without the radio bearer being set-up. Later during the commit phase the radio bearer will be set-up.

Phase 1 and phase 3 are SIP message exchanges. These phases are not dependent on the access technology itself and therefore they can be applied to UMTS without modifications. Nevertheless, they introduce additional signalling over the air interface as SIP is carried end-to-end.

This counterbalances the resource efficiency benefits of allocating resources only after the caller has picked up. Additionally, the post pick-up delay may be large as the resource allocation procedure in UMTS can be time consuming (phase 4).

5 Conclusion

In this article we presented an overview of the QoS framework of the UMTS with a focus on the packet domain. The QoS is provisioned per session through the establishment and maintenance of bearers at the different layers of UMTS. The QoS profile of a session includes parameters describing the characteristics of the traffic flow so as to enable efficient resource allocation.

We also described the ongoing work and assumptions regarding the provisioning of an end-to-end bearer service. Among other issues the mapping of IP QoS to UMTS QoS is a challenging task and the co-ordination of the call control and the end-to-end bearer control needs to be improved.

6 References

1 3GPP. *QoS Concept and Architecture, v 3.5.0.* (TS 23.107)

2 3GPP. *End to End QoS Concept and Architecture, v 1.5.0.* (TS 23.207)

3 3GPP. *IP Multimedia(IM) Subsystem-stage 2, v 1.7.0.* (TS 23.228)

4 3GPP. *Handover for real time Services from the PS domain, v 0.3.0.* (TR 25.936)

5 3GPP. *General Packet Radio Service (GPRS) Service description; Stage 2, v 3.6.0.* (TS 23.060)

6 3GPP. *Network Architecture, v 5.1.0.* (TS 23.002)

7 3GPP. *Architecture Requirements for release 99,v 5.0.0.* (TS 23.121)

8 3GPP. *UTRAN Overall description, v 3.5.0.* (TS 25.401)

9 3GPP. *UTRAN Iu interface: General Aspects and Principals, v3.3.0.* (TS 25.410)

10 3GPP. *UTRAN Iu interface: layer 1,v3.3.0.* (TS 25.411)

11 3GPP. *UTRAN Iu Interface: signalling Transport, v3.4.0.* (TS 25.412)

12 3GPP. *UTRAN interface: Data transport and Transport Signalling, v3.6.0.* (TS 25.414)

13 3GPP. *UTRAN Iu interface: User plane protocol, v3.5.0.* (TS 25.415)

14 3GPP. *UTRAN Iur interface General aspects and Principals, v3.2.0.* (TS 25.420)

15 3GPP. *UTRAN Iur Interface: layer 1,v3.0.0.* (TS 25.421)

16 3GPP. *UTRAN Iur Interface: Signalling Transport, v3.5.0.* (TS 25.422)

17 3GPP. *UTRAN iub interface: General Aspects and Principals, v3.4.0.* (TS 25.430)

18 3GPP. *UTRAN Iub interface, Layer 1,v3.0.0.* (TS 25.431)

19 3GPP. *UTRAN Iub Interface: Signalling transport, v3.1.0.* (TS 25.432)

19 3GPP. *UTRAN Iub interface: Data transport and Transport signalling for CCH Data streams ,v3.4.0.* (TS 25.434)

20 3GPP. *UTRAN Iub interface: User plane protocol for CCH Data streams, v3.5.0.* (TS 25.435)

21 3GPP. *Delay budget within the access stratum, v 4.0.0.* (TR 25.853)

22 3GPP. *Radio Resource Management and Strategies.* (TS 25 922)

23 Holma, H, Toskala, A. *WCDMA for UMTS, Radio access for Third Generation Mobile Communications.* Chichester, Wiley, 2000.

24 PacketCable Distributed Call Signalling Specification, PKT-SP-DCS-D17-03-000428, PacketCable Draft internal document.

25 PacketCable Dynamic Quality of Service Specification, PKT-SP-I01-991201. Available at http://www.packetcable.com/.

Optical Network Functionality: From “Dumb Fat Pipes” to Bright Networking

EVI ZOUGANELI

Optical technology has experienced an explosive growth in the past years that has enabled multi-terabit optical transmission over several hundreds of kilometres. Yet the potential of optical networking is far from being exploited. Optical network functionality may be the answer to efficient and reliable data-centric networks, a technology that complements IP. This article aims at giving an overview of the driving forces behind optical networking, the potential offered by it, the challenges encountered, and the current state-of-the-art.

Evi Zouganeli (38) studied Applied Physics at the Univ. of Patras, Greece, and after earning the national postgraduate scholarship in 1986, she obtained the MSc in Telecommunications (1988) and PhD in Optoelectronics (1992), both from University College London, UK. She is currently Senior Research Scientist at Telenor R&D with interests focussing on optical network architectures. She joined Telenor R&D in 1994 and has since worked on high capacity optical transmission, optical networking and migration scenarios, as well as strategies for the upgrading of the Norwegian network – both for Telenor BUs and in European collaboration projects. She is member of a number of international Technical Committees. Prior to joining Telenor, she was with the Federal Institute of Technology, Zürich, Switzerland, where she worked with optical switching and high capacity optical networks.

evi.zouganeli@telenor.com

1 Introduction

Telecommunications as we have known it, is being literally transformed with the convergence of voice services with data and multimedia services, mobility requirements, and the presence of several competing players in the market. The industrial age is allegedly over, the world is going digital, global and on the net, and we have entered the information era where new paradigms are dominating corporate, social as well as private life. Despite some ups and downs in the forecasts, the fact remains that data traffic has increased dramatically in the past years and has now surpassed voice traffic in volume, signifying a soft transition to the new economy. But how do our networks catch up with this (r)evolution?

Today’s SONET/SDH network infrastructure provides a guaranteed level of performance and reliability for voice calls and leased lines. Existing networks have been designed for telephony and are thus adequate for handling static traffic patterns but rather inefficient in handling the new traffic patterns that are dominated by data traffic. In contrast to traffic generated by telephony, data traffic patterns are more unpredictable, asymmetric in terms of load distribution, and bursty in nature. Convergence in the applications front, the potential for more sophisticated services and the requirement for tailored billing mechanisms in view of the expanding competition in the liberalised telecom market, create a positive feedback loop that further strengthens the requirement for more dynamic and flexible networks. In addition, because of the dramatic increase in the capacity carried by each cable (of the order of Tbit/s), it is mandatory to have reliable and fast ways to restore the network in case of fibre cut or other failure and to be able to prioritise traffic depending on the carried service.

All in all, there is clearly a need to realise networks that can adapt to changing traffic requirements and can make a good use of the network resources, with fast automatic reconfiguration, efficient traffic engineering, and service differentiation. These will allow a rationalised – i.e. economical – use of the network, increase the synergy between the different network platforms and enable the introduction of new value-added services.

At the physical layer, the tremendous increase in network capacity has been facilitated by technological advances in optical transmission systems, i.e. the deployment of dense wavelength division multiplexing (DWDM). Optical technology has followed a rather explosive growth curve in the past five years or so, primarily driven by the growth in Internet traffic, and – not least – future growth expectations. Optical transmission is widely used in most parts of the network worldwide, namely from transatlantic and pan-European connections, to national connections between cities as well as within cities. Metropolitan area networks are being built and becoming predominantly fibre-based as many businesses require high-bandwidth connections, and fibre installations are taking off substantially also in the residential customer access area.

The role optical technology has played so far has been that of a “dumb fat pipe” – as it has been described with a certain touch of endearment. The network functionality potential of optical technology is still largely unexploited and networks are today built using layer upon layer with duplicated functionality. This is rather uneconomical both in terms of cost and in terms of time consumption, hence there is strong consensus in the engineering community that network architectures need to be rationalised, piles of unnecessary equipment removed, and some layers collapsed. What is not quite clear yet is exactly how this ought to be realised – what architecture is most efficient, future proof, and feasible – as well as what role optical technology will play to best facilitate network evolution.

2 Optical Network Functionality

Though relatively immature, the technology is available to enable optical networks that use DWDM not only as a means to increase capacity in the fibre, but rather as a means to provide direct connectivity and intelligent re-configurability in the network. DWDM optical channels are distinguishable as based upon the wavelength of the channel. Most optical components are actually inherently wavelength dependent, a fact that may add a complication to transmission systems sometimes but it also provides an explicit way to identify and route/process a signal without needing to read its content – transparently. It is this quality of optical signals that gives optical networking a huge competitive advantage against other technologies – electronics – namely because it makes it inherently scalable in terms of speed as well as bitrate-, format- and protocol-blind.

Although the field of optical networks is in full expansion, very little optical network functionality has actually been implemented in the network as yet. DWDM is used to increase the bandwidth in the fibre and provide point-to-point connections. Optical add-drop multiplexers (OADMs) are available and can provide direct optical connections between end nodes, bypassing intermediate nodes and thus eliminating unnecessary electronic processing. Yet the OADMs that are installed today are primarily fixed wavelength, which means that a predetermined set of wavelengths may be tapped out at a certain node but no programmable re-configuration is possible. This minimises the potential of these in a network context exactly because it makes them inaccessible for the management system. Optical channels are thus still providing point-to-point connections in these configurations. Commercial solutions with a management system that allows point-and-click provisioning at optical channel level are just emerging, however, automatic provisioning via signalling directly from a client (network) is still not available.

In terms of network topology, optical networks will consist of a multiple of sub-networks as dictated by administrative geographical and technological factors. These will need to be interconnected by optical links in an arbitrary topology, i.e. in a mesh topology as the physical network is in the general case a mesh network. Mesh topologies make the best use of the available bandwidth, facilitate load balancing in the network, as well as provide multiple and short restoration paths. In order to realise optical mesh networks, optical cross-connects (OXC) are required, i.e. programmable switching matrices with several input and output fibres that can direct optical channels from any fibre input to any fibre output [1].

OXCs are just emerging – therefore currently expensive – network elements (NEs) and some time will be needed before the winning technologies are identified and they become mature, widely used systems. OXCs allow switching of channels to different directions in a mesh network and give full flexibility with regard to physical path selection within the network, thus enabling re-configurable optical networks. End-to-end optical channel (OCh) connections can be established between two end-nodes by reconfiguring the OXCs along a certain path that connects these nodes. “Point-and-click” provisioning of OCh’s can be achieved this way and protection or restoration can be carried out fast in the optical domain, when required. Bandwidth can be allocated on-demand or “created” at the parts of the network where it is required. Note that an OCh is not necessarily all-optical along the end-to-end link; neither is it necessarily one single wavelength along the whole path.

With channel speeds at 10 Gbit/s being the state-of-the-art today, and 40 Gbit/s emerging, a channel count of over 100 channels per fibre and transmission distances of over 3000 km without opto-electronic conversion, it has become economical to perform switching/routing functions in the optical domain – in any case in the core network. IP routers with multiple optical 10 Gbit interfaces are the state-of-the-art today. Handling such large volumes of traffic electronically packet by packet creates huge bottlenecks and impedes total network throughput. Additionally, opto-electronic (O/E/O) conversions are expensive especially at such bit-rates and should be avoided as much as possible. It is estimated that over two thirds of all traffic arriving at a node is passing-through traffic in the core network. Therefore bypassing nodes optically brings significant cost savings as well as simplifies and speeds up network functions.

Optics does have its shortcomings: there is no good optical memory as yet and neither a good optical buffer. These characteristics or limitations determine the way we regard network architectures today: processing of information as such is done electronically. The notion is to move the electronic processing to the edge of the network as much as possible and carry out optical network functions in-between.

Ultimately, provided some of the limitations are overcome, signals may be processed optically. A step before real optical data processing is optical packet switching [2, 3] or optical burst switching where packets or (respectively) streams of packets are encapsulated in an optical “container” that is identifiable by its wavelength. This article is limited to the shorter term, thus to optical circuit switched networks, so that we will not refer to optical packet or optical burst switching here. However, the network architectures that are discussed in the following are also applicable to optical packet switched networks.

3 Network Architectures

With regard to architecture alternatives for IP over optical networks, the determining aspect is whether and to what degree the control plane of the optical network will be integrated with IP or independent of it. The IP and optical control planes can in other words be loosely or tightly coupled in terms of, firstly, the details of the optical network topology, resources, and routing information that is revealed to the IP layer, and secondly, the degree of control IP routers have on optical network elements and thus the degree to which they can determine the exact paths through this optical network. Three architecture options can be identified from this point of view:

The overlay model: In this architecture option the optical network has full control over its network resources by means of a fully independent optical control plane (Figure 1). Communication with its clients, among these also IP, is done via a well-defined User-Network Interface (UNI) at the edge of the network where only signalling information is exchanged. The client networks request a connection between two edge nodes, requesting also certain quality related characteristics for this connection. These characteristics do not only regard bandwidth but also e.g. delay, jitter, degree of protection, etc. The client networks have otherwise no control over the exact routing and priority received within the optical network.

This "bottom-up" model is very popular with vendors that have long tradition in optical systems, heavy optical expertise, and only recent experience with IP (hardware in any case). IP-centric vendors have also opted for this model in their first phase products as this model is feasible in the short term. Here an "intelligent optical network" carries out part of the network functionality. Another advantage of this model is that it is a multi-client solution, which can accommodate technologies other than IP that many operators will need to relate to, at least for a while. Separating the two control planes implies also that the two parts may evolve, be adapted, and be optimised independently, which is a good future-proof policy. The optical network provides here a universal platform that is not tied to one specific protocol but is open to any future new-comers. This aspect is especially important at this stage when optical technology is experiencing intense growth and is therefore exposed to large changes. The disadvantage of the overlay model, on the other hand, is that it requires the creation of a new control plane that to an extent duplicates functionality and may introduce delays – repeating that is the old problem with layered networks. It can be expected that as IP gradually displaces alternative technologies, the overlay architecture will at some point become an anachronism.

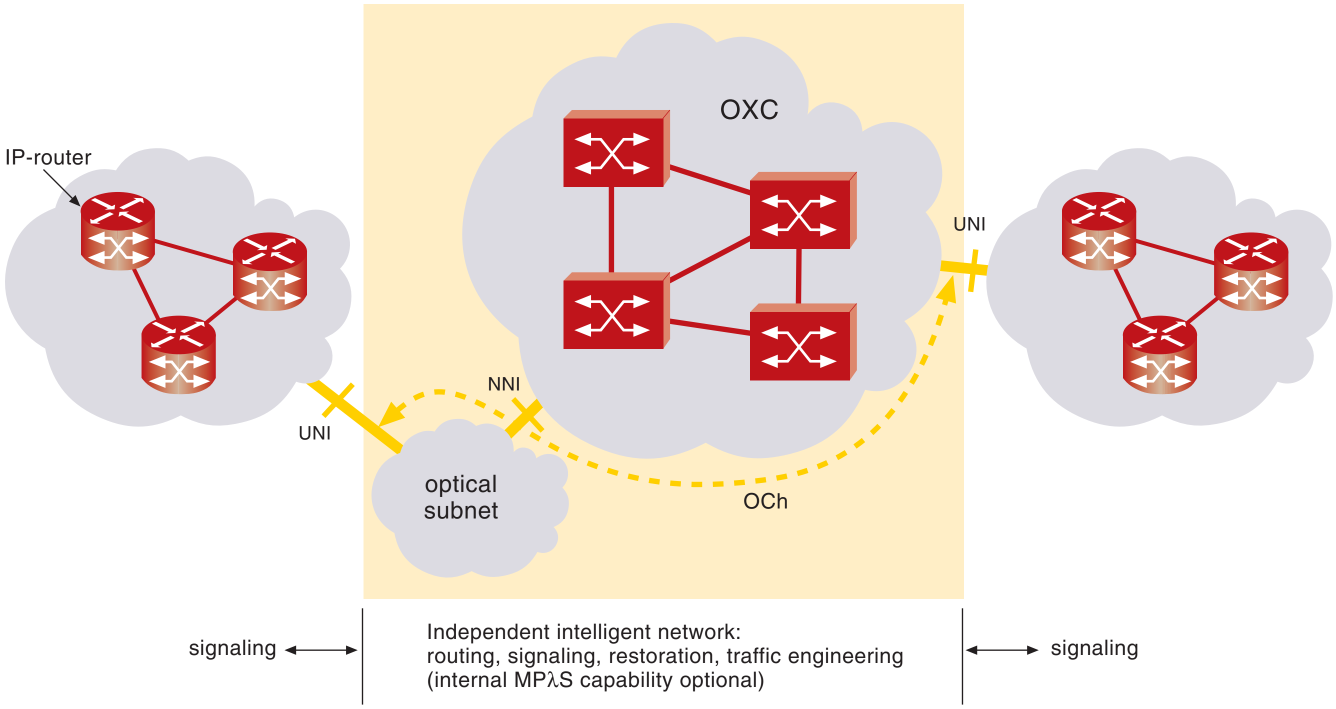

OXC
IP-router
UNI
UNI
NNI
optical subnet
OCh
signaling
Independent intelligent network:
routing, signaling, restoration, traffic engineering
(internal MPλS capability optional)
signaling

Figure 1 The overlay model. An intelligent optical network has full control over its network resources and offers end-to-end wavelength services to client networks via a well-defined User Network Interface

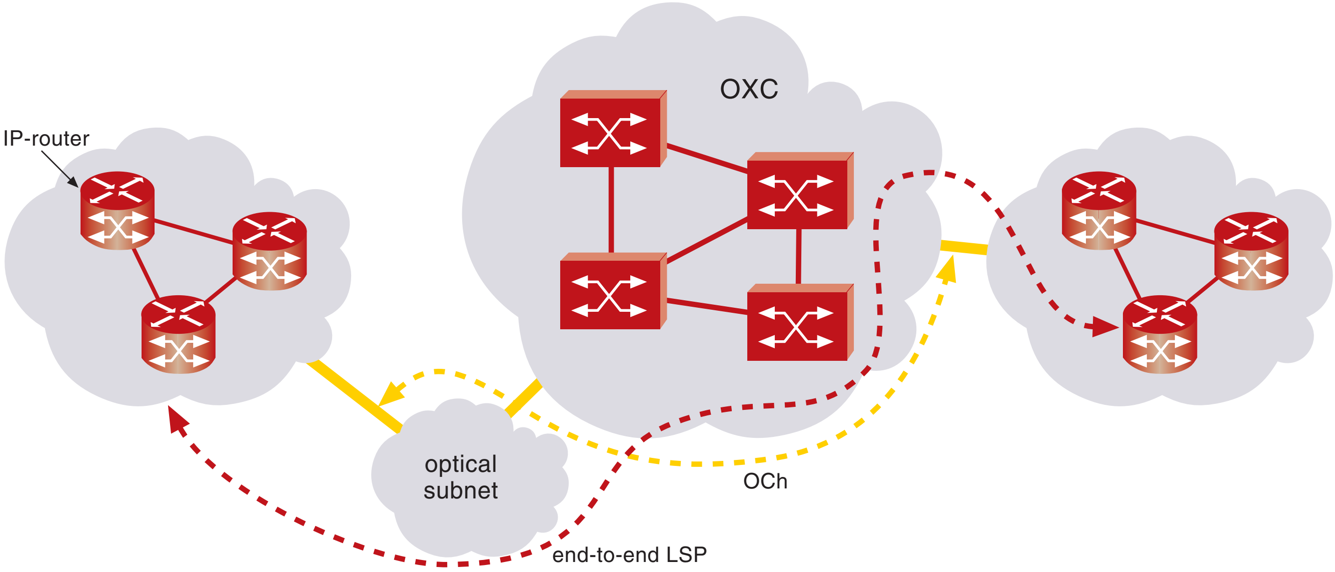


Figure 2 The Peer-to-Peer model. IP routers and optical network elements (OXCs) are peers in a merged network that is managed seamlessly in a unified way

The peer model: In this architecture the control planes of the optical network and IP are fully integrated such that IP routers and optical switching nodes (OXCs) are peers. The two networks are merged into a new integrated network that is managed in a unified way. The optical network topology is fully visible to routers. A single protocol is run through all domains and establishes paths through all network elements in a seamless manner (Figure 2).

This "top-down" approach is primarily the view of IP experts and equipment suppliers that have good expertise in IP but little experience with optical technology. The view is to keep optical "dumb and fat" as it has been up to now and rely on IP intelligence to run the network. The advantage of this architecture stems exactly from the fact that it is IP-centric: optimised routes may be found through the network taking into account all factors, also physical factors. The architecture is scalable, functionality is not duplicated and conflicts between several control planes do not arise. On the other hand, this architecture demands that information regarding the optical network elements is advertised to routers, resulting in excessive information flows within the network and an overblown control system. Thorough adaptations of the routers are required in that routing information that is specific to optical networks needs to be incorporated in the protocols. The degree to which this creates conflicts is still unclear. Both software and hardware adaptations can in any case be required that may not be easy to implement in the short term. Finally, this architecture is not inherently multi-client, an aspect that may be important for some operators (e.g. incumbents). If the amount of non-IP traffic is relatively small, this traffic may be carried over IP. However, leasing of "dark" capacity lines is not facilitated by this architecture. Despite its drawbacks, the peer-to-peer model can be expected to be the architecture to be adopted in the longer term if IP indeed dominates the scene.

The augmented model: This architecture can be a whole range of solutions that lie in the area between the peer and the overlay model. Here the network can be seen as comprising separate IP and optical domains, where each one is using a separate instance of an interior routing protocol. Some reachability information is exchanged between these domains but the topology of the optical network is by and large opaque to the client network. This option may be a good compromise in that it is more feasible than the peer solution and at the same time less rigid and more efficient than the overlay model. It can be argued that this solution combines the best of two worlds because it limits the amount of info exchange within the network and at the same time it may allow the delivery of wavelength services, i.e. trading of pure end-to-end bandwidth, that circumvent the IP layer.

The above three architectures were initially seen as rival solutions. Lately it appears that as the realities of pros and cons for all options are becoming more evident, the three solutions are seen as possible evolution stages down one single path. According to this scenario, networks will first be based on the overlay model, proceed to an augmented model with enhanced signalling as well as routing information exchange between different domains, and finally – assuming IP becomes the transport protocol – move to the peer model with a full integration between the optical and the IP plane. This scenario may be challenged if good solutions based on the aug-

MPLS	Optical Wavelength Routed Network
Label Switched Path (LSP)	Optical Channel (OCh)
Label Switching Router (LSR)	Optical Cross-Connect (OXC)
Selection of Labels	Selection of λ's (possibly in combination with OXC ports)

Table 1 Analogy between MPLS and Optical Wavelength Routed Network

mented model appear timely enough. Also, solutions based on the peer model may be implemented right from the start by newly established Internet Service Providers (ISPs) that also have ownership of the physical infrastructure, or in the cases where the amount of non-IP traffic is relatively small.

4 Control System for Optical Networks: Generalised Multi-protocol Label Switching (GMPLS)

Multi-protocol label switching (MPLS) was launched only a couple of years ago but has now become a fundamentally important technology in the Internet. Several of the largest Internet service providers employ MPLS in their networks, virtual private network (VPN) services based on MPLS are now available and the majority of high-end routers now support MPLS [4]. MPLS is an IP-centric protocol at the same time that it is independent of the IP framework. It is a standardised solution to place the handling of traffic as much as possible down to Layer 2, i.e. perform "switching" instead of "routing". This circumvents the major of the shortcomings of IP as it simplifies routing processes, provides efficient and reliable handling of larger traffic volumes, as well as enabling traffic engineering, faster restoration, and easier QoS handling.

Packets are classified in flows (Forwarding Equivalence Class, FEC) where the same routing decision is applied. Label switching comprises mainly the allocation of a stack of labels to each packet (flow) where each label refers to a different network layer within a hierarchical network. The label can be "deciphered" to a forwarding port by each router according to a frequently updated routing table and a new label attached giving forwarding directions to the next router. Routing and signalling protocols are a part of the process in order to discover the network topology, place and respond to requests, reserve and establish paths. MPLS is presented in detail in a separate article in this edition of *Telektronikk,* so that we can close this short introduction to it here.

The main principle behind MPLS has a lot in common with inherent features of wavelength routed optical networks. Indeed, optical network elements with interfaces that can recognise wavelength, such as optical add-drop multiplexers, de-multiplexers and cross-connects – can direct the optical signal based on its wavelength without the need for reading the content of the signal. This is clearly a form of label switching [5]. The optical label is in a different domain (optical) than the signal itself (electrical). This aspect makes wavelength the perfect label since no overhead is added to the signal. The analogy between MPLS networks and optical wavelength-routed networks is shown in Table 1.

Responding to the clear mismatch between the transmission capacity of the fibre and the processing capacity of routers, the Internet Engineering Task Force (IETF) has more recently proposed and is developing an extension of MPLS – Generalized MPLS (GMPLS) – to the time and the optical domain. The aim is to achieve a more flexible labelling and forwarding mechanism that uses a generalised label, which is applicable to a variety of technologies and networks. For IP routers the labels designate principally input and output ports. For an OXC they designate input and output ports as well as wavelength, or band of wavelengths. The hierarchy of different labels in GMPLS is schematically shown in Figure 3.

The main aims of GMPLS are to [6]:

i provide a framework for real time provisioning of optical channels;

ii adopt optical technology and encompass the development and deployment of a new class of programmable OXCs;

iii allow the use of uniform semantics for network control in hybrid networks that consist of both OXCs and label switching routers.

Using GMPLS, an end-to-end optical channel can be established between two end-nodes by choosing a) a physical path that connects these

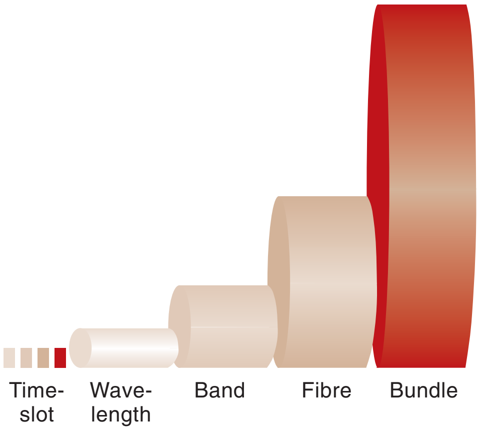


Figure 3 A schematic of the hierarchy of labels in GMPLS

nodes, and b) the wavelengths used in each fibre section of this path. The OXCs connecting the fibre sections are then configured to direct the signal entering from a certain port to the right output port as well as assigning the signal a new wavelength, if required, so that it can be transmitted to the next OXC down the chosen path. The process is repeated until the signal has reached its destination. A large volume (flow) of packets can thus be directed to their destination node by means of an optical label, which consists of the optical wavelength and the output port of the OXC, given a certain input port. By employing optical technology in this way and performing a single routing decision for a large volume of packets, the whole routing process is carried out much more efficiently and the throughput of the IP network is dramatically improved.

At the same time, since MPLS was not originally created with optical technology in mind, there are certain aspects of it that either cannot be realised in optical networks or are difficult to realise in optical networks – and therefore ought best to be avoided. As a result, the adaptations and extensions of MPLS protocols to GMPLS may not be enough when optimising optical networks that implement GMPLS: a change of thought may be needed in addition as compared with the implementation of MPLS. This stems from the fact that routing in optical networks cannot be dissociated from the physical layer the way MPLS functions can take place entirely at higher network layers. MPLS has been described as Layer 2.5. GMPLS is still performing MPLS functionality, i.e. in interplay with Layer 3, but at the same time it is at the doorstep of Layer 1 – bridging exactly the two. Some of the standard MPLS processes that are not particularly facilitated by optical technology, are listed below:

- Label merging is not straightforward to realise optically and it may require a smart combination with electronic techniques.

- Label stacking, pushing and popping appears rather complex and expensive to realise optically – if at all possible; it will probably require a smart combination with electronic techniques. Intense R&D work is taking place in this area.

- Label swapping can be carried out optically (e.g. by wavelength conversion) but it is expensive and should be minimised in optical networks.

Provisioning in optical networks: routing and signalling

Extensions to MPLS signalling and routing protocols are being developed by the IETF in order to include the specific requirements of optical technology.

Routing in optical networks consists of the routing problem – with or without some form of constraint – and the wavelength assignment problem. The two are in the general case not dissociated.

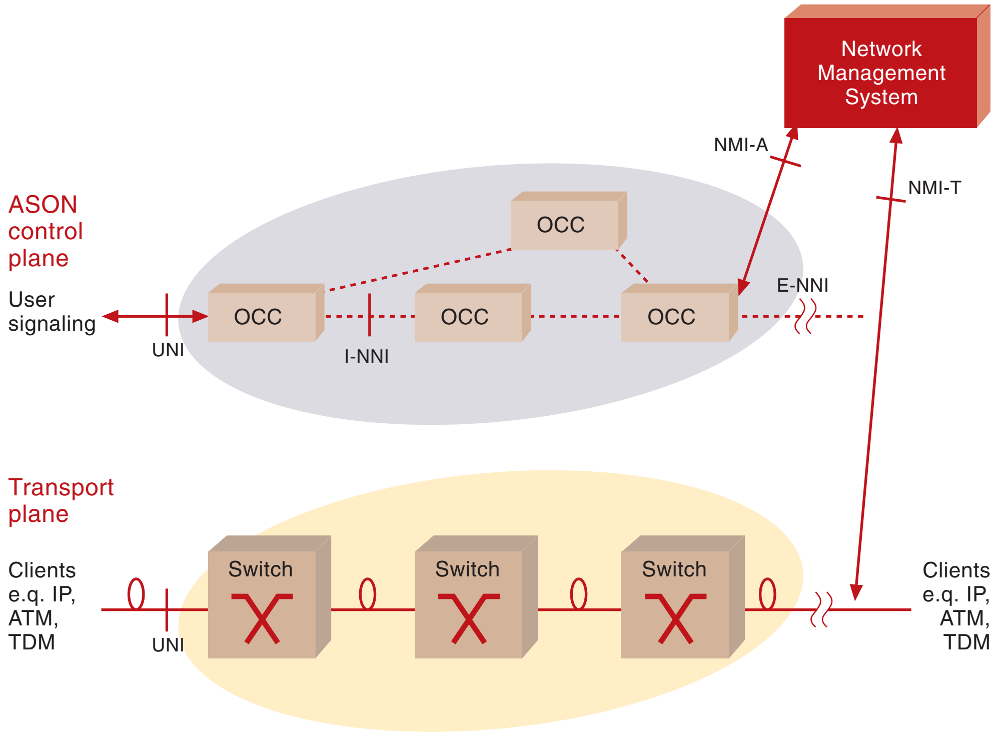
One of the basic rules in DWDM is that two signals in the same fibre cannot have the same wavelength. When calculating the best path for a connection, path length, number of hops, and bandwidth availability are not the only factors that need to be taken into account. Wavelength allocation must be an integral part of the routing algorithm and this needs to minimise wavelength conversions in the general case – if these are at all allowed. The main aspect in routing algorithms under development concerns minimising congestion of OChs and eliminating violations of the "unique wavelength" rule in DWDM. Additionally, physical characteristics of the optical link – such as signal quality, noise, etc. – may need to be incorporated in the protocols such that the right routes are chosen depending upon class of service (CoS) in order to facilitate traffic engineering. Optical routing algorithms often involve some type of constraint based routing.

Firstly, neighbour discovery procedures are required in the network to identify the optical nodes, their connections, etc. This can be based on existing protocols such as NDP [7]. Also a link state update is required, which – as the name implies – provides an update of the status of all links in a sub-network. If a fully distributed approach is chosen for path establishment, then the link update needs to take place at all OXCs. An IP link state protocol such as OSPF may be adapted to carry out this function [7].

Route computation is based on the information revealed by the link state update. In a distributed implementation the request for path establishment is forwarded to the OXC at the ingress point that is then responsible for the computation and the establishment of the path. Protection and/or restoration routes may be optimised globally and off-line or locally at the OXC and real-time [8, 9].

Path establishment is again carried out based on IP models. The MPLS architecture for IP networks implements either RSVP or Constraint Routed LDP (CR-LDP). These existing protocols for establishing label switched paths are being extended to encompass optical technology.

Figure 4 Automatically Switched Optical Network after ITU-T



5 Automatically Switched Optical Networks (ASON)

ITU-T has been active in the standardisation of the Automatically Switched Optical Network (ASON) [10], which is an optical transport network with an independent control plane (Figure 4). ASON is in other words based on the overlay network architecture and consists of three main components: the control plane, the transport plane and the network management plane. ASON is a multi-client network that can offer connection services to client networks (IP, ATM, SDH, etc.). These can be set up following a request via the network management system (NMS) or following direct signalling exchange between the client network and ASON. To that end, work has been carried out in parallel by the Optical Internetworking Forum (OIF) in order to define and implement the User Network Interface (UNI). All communication with client networks in an ASON is done via this interface. This interface carries all signalling information exchanged between ASON and its clients as well as the actual signals transported by the network (via the physical layer part of the interface). No routing information is exchanged through the UNI. The work on the UNI carried out by the signalling group of OIF has recently been co-ordinated with the IETF. A first version of the optical UNI was used to demonstrate interoperability testing for equipment from different vendors at Supercomm 2001, as a result of the joint efforts of 25 vendors within OIF.

Optical Connection Services

ASON provides end-to-end OCh connections to its clients with a certain QoS, as agreed via service level agreements (SLA) with the client. These connections can be static, established via the management system, or dynamic. The following three types of OCh services can be provided:

- **Permanent OCh connection**
  This provides a permanent end-to-end connection at optical channel granularity. The service is requested by the client via the NMS and activated by the NMS.

- **Soft-permanent OCh connection**
  This provides an end-to-end OCh over a certain period of time. The service is requested by the client via the NMS but the connection can be established using signalling in the control plane.

- **Automatically switched OCh connection**
  This provides an end-to-end optical channel connection activated by direct signalling from the client network. Establishment and tear down of this service are handled automatically by the ASON control plane and the client is notified accordingly. The NMS is periodically updated.

The Control Plane

The control plane of ASON is what distinguishes it from a simple optical transport network (OTN). ASON acquires in other words network

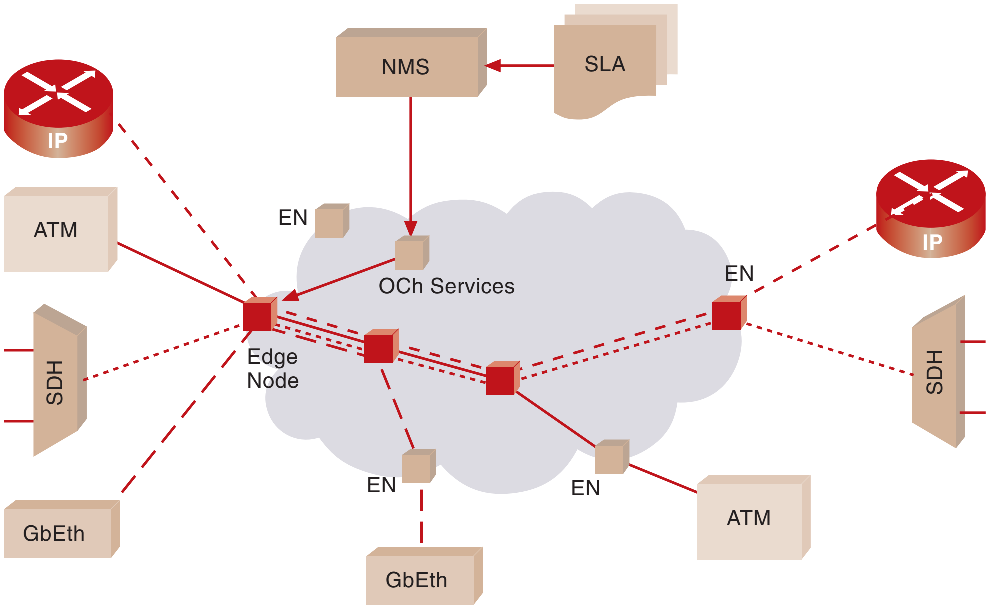


Figure 5 Multi-client end-to-end wavelength services offered by ASON – here an example with wavelength allocation performed separately for each client

intelligence. The functionality required by the control plane is based on IP models. Thus the functionality carried out by the control plane here includes service invocation via the UNI, neighbour discovery, status updating, as well as routing of optical channels, wavelength allocation and path establishment. These can be based on GMPLS as presented in the previous section. Indeed, both effective development when adapting existing well-proven protocols, and easier migration scenarios towards future solutions, justify this choice.

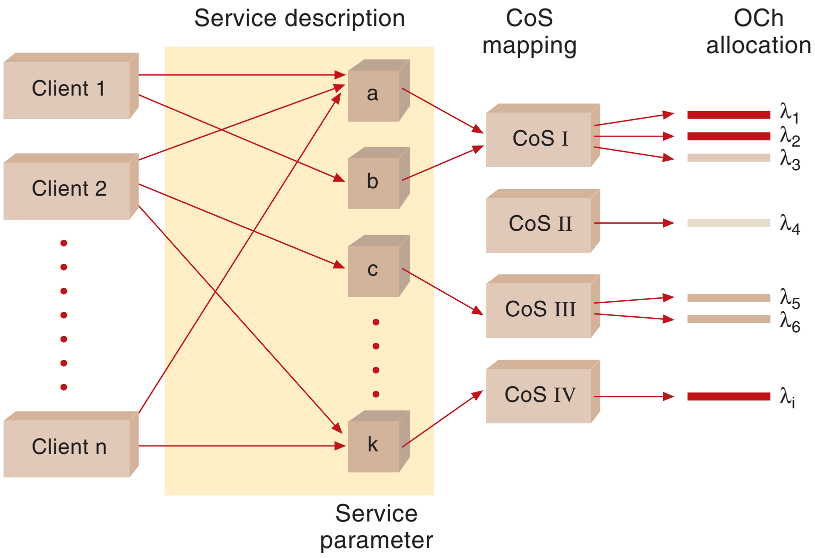
The control plane comprises local Optical Connection Controllers (OCC) at each node that exchanges information with the optical network via the Connection Controller Interface (CCI). The OCC instructs the local optical switch to reconfigure via the CCI, which also carries topology updates from the node. Communication between OCCs is done via the Network Network Interface (NNI). This can be an internal network interface (I-NNI) when it carries routing/signalling messages within a single ASON administrative domain (e.g. a single operator), or an external network interface (E-NNI) when it connects two separate administrative domains. The main difference between the two is that no topology information is carried through the E-NNI; neither is resource control possible through this interface. Finally, the control plane communicates with the network management system via the NNI-A and NNI-T interfaces.

Signalling information within the control plane can be embedded information but is best conveyed using a separate signalling channel. This signalling channel can be *out-of-fibre*, i.e. going through an altogether separate network, or *in-fibre*. In the latter case the signalling channel can be *in-band*, i.e. within the range of the (standardised) optical channels that are used for the transmission of traffic, or *out-of-band*. The signalling channel requires a dedicated resilience strategy. GMPLS-based signalling protocols are envisaged used in ASON as well as through the UNI and a lot of development work is being carried out in this area.

Classes of Service

ASON can provide service differentiation as based upon a set of parameters such as priority, resilience, delay, jitter, etc., following IP resource handling models [12]. Figure 5 shows an example of an ASON providing end-to-end connections arranged by client. Optical channels are then allocated separately to each client, according to a set of classes of service (CoS) as shown in Figure 6. This would place traffic

Figure 6 Service requirements are mapped in Classes of Services, followed by route computation (not shown) and wavelength allocation



grooming outside ASON. Alternatively, bandwidth allocation can be carried out using a wavelength allocation scheme that can allow mixing of a number of clients together. Network segment – e.g. metro versus core networks – may be the most decisive factor when choosing among different traffic grooming paradigms.

Automatically switched optical networks can allocate resources on demand, accommodate lower priority traffic where and when there is bandwidth available, and drop this traffic if necessary in case of congestion or in case higher priority traffic needs to be restored because of fibre break or other failure. The dynamic provisioning, bandwidth efficiency, traffic engineering and fast restoration capabilities of ASON make it a very powerful multi-client platform that opens new possibilities for service integration, the introduction of new services, new pricing paradigms and business models.

6 Summary

Optical transmission has experienced a dramatic growth in the past years with the advent of DWDM that has enabled multi-terabit optical transmission over distances of several hundreds of kilometres. Yet the potential of optical technology is still far from being exploited. Optical network functionality in wavelength-routed networks may well be the key to high capacity intelligent networks that utilise their resources in an efficient way and can provide a range of differentiated services. Optical switching provides an economical way to handle large amounts of traffic and to build reliable networks. It leads to a dramatic reduction of the required processing capacity and to rationalised network architectures without duplication of functionality and expensive superfluous interfaces. The on-going intense standardisation and development work in the past years can mean that such networks are not all that far from being a reality; they are in fact leaving the labs and entering the market as we speak. Although the winning technologies are not yet fully identified and the detailed logistics of networks and architectures are still to be finalised, one reality has clearly emerged: Optical switching will be an integral and determining part of next generation networks.

7 References

1 Jackman, N A et al. 1999. Optical Cross Connects for Optical Networking. *Bell Labs Technical Journal,* 4 (1), 262–281.

2 Hunder, D K, Andonovic, I. 2000. Approaches to Optical Internet Packet Switching. *IEEE Communications Magazine,* 38 (9), 116–122.

3 O'Mahoney, M J et al. 2001. The application of optical packet switching in future communication networks. *IEEE Communications Magazine,* 39 (3), 128–135.

4 Davie, B, Rekchter, Y. *MPLS: Technology and Applications.* San Francisco, Morgan Kaufmann, 2000.

5 Ghani, N. 2000. Lambda Labeling: A Framework for IP-over-WDM using MPLS. *Optical Networks Magazine,* 1 (2), 45–57.

6 IETF. Awduche, D O et al. 2001. *Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Cross Connects.* (Internet Draft.)

7 Bala, R et al. 2000. IP over Optical Networks: Architectural Aspects. *IEEE Communications Magazine,* 38 (9), 94–102.

8 Ali, M A et al. 2001. Architectural options for the next-generation networking paradigm: Is Optical Internet the Answer? *Photonic Network Communications,* 3 (1/2), 7–21.

9 Listanti, M et al. 2000. Architectural and Technological issues for Future Optical Internet Networks. *IEEE Communications Magazine,* 38 (9), 82–92.

10 EURESCOM. 2001. *Deliverables, EURESCOM project P1012 FASHION: Flexible Automatically Switched Client Independent Optical Networks.* [online] – URL: www.eurescom.de

11 OIF. 2001. *Supercomm 2001 OIF UNI Demonstration White Paper.* [online] – URL: www.oiforum.com

12 Svinnset, I et al. *Resource Handling in IP Networks.* Kjeller, Telenor R&D, 2001. (R&D R 5/2001.)

Abbreviations

A

AAA	Authentication Authorisation Accounting
AAAARCH	Authentication Authorisation Accounting ARCHitecture research group
AAL	ATM Adaptation Layer
AC	Alternating Current Application Categories
ACK	Acknowledgement
ACS	Admission Control Service
ADSL	Asymmetric Digital Subscriber Line
AF	Assured Forwarding
AH	Authentication Header
AMR	Adaptive MultiRate
API	Application Programming Interface
ARIB	Association of Radio Industries and Businesses
ARPA	Advanced Research Projects Agency
AS	Autonomous System
ASCII	American Standard Code and Information Interchange
ASI	Application Specific Information
ASIC	Application-Specific Integrated Circuit
ASM	Application Specific Module
ASN.1	Abstract Syntax Notation no.1
ASON	Automatically Switched Optical Network
ASP	Application Service Provider
ATM	Asynchronous Transfer Mode

B

BA	Behaviour Aggregate
BB	Bandwidth Broker
BBE	Better than Best Effort
BE	Best Effort
BER	Bit Error Ratio
BGP	Border Gateway Protocol
BGP-4	Border Gateway Protocol, version 4
BICC	Bearer-Independent Call Control
B-ISDN	Broadband Integrated Services Digital Network
BLA	Business Level Agreement
BR	Border Router
BS	Bearer Service

C

CAC	Call Admission Control Connection Admission Control
CAR	Committed Access Rate
CATV	Cable Television
CBC	Cipher Block Chaining
CBR	Constraint-Based Routing
CBS	Committed Burst Size
CBT	Core Based Tree
CBWFQ	Class-Based Weighted Fair Queueing
CC	Connection Count
CCI	Connection Controller Interface
CCITT	International Telegraph and Telephone Consultative Committee (now: ITU-T)
CDF	Complementary Distribution Function
CDMA	Code Division Multiple Access
CDR	Committed Data Rate
CE	Congestion Experienced (part of ECN mechanism) Customer Edge
CI	Configuration Interface
CID	Class Identity
CIM	Common Information Model
CIR	Committed Information Rate
CIV	Common Information Viewpoint
CL	Controlled Load (Intserv class)
CLP	Cell Loss Priority
CMIP	Common Management Information
CN	Core Network
COA	Care-Of-Address
COPS	Common Open Policy Service
CORBA	Common Object Request Broker Architecture
CoS	Class of Service
CP	Connection Point
CPE	Customer Premises Equipment
CPG	Connection Point Group
CPODA	Contention Priority Oriented Demand Access
CPU	Central Processing Unit
CR	Core Router
CR-LDP	Constraint-Based Label Distribution Protocol
CRT	Cathode Ray Terminal
CS	Circuit Switched Class Selector (Diffserv class)
CSCF	Call State Control Function
CS-MGW	Circuit Switched Media Gateway
CSPF	Constrained Shortest Path First

CSRC	Contribution Source identifier
CTI	Computer Telephony Integration
CTPG	Connection Termination Point Group
CTSS	Computer Time-Sharing System
CU	Current Unused
cwnd	congestion window
CWTS	China Wireless Telecommunication Standard Group

D

DARPA	Defence Advanced Research Projects Agency
DC	Default Class
DCS	Distributed Call Signalling
DE	Default class
DEC	Decision Digital Equipment Corporation
DEN	Directory Enabled Network
DES	Data Encryption Standard
DF	Distribution Function
DHCP	Dynamic Host Configuration Protocol
DI	Data Interface
DiffServ	Differentiated Services
Dir	Directionality
DMTF	Distributed Management Task Force
DNS	Domain Name System
DQoS	Dynamic Quality of Service
DS	Differentiated Services
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
DSP	Digital Signal Processor
DTMF	Dual Tone MultiFrequency
DVMRP	Distance Vector Multicast Routing Protocol
DWDM	Dense Wavelength Division Multiplexing
DWFQ	Distributed Weighted Fair Queueing

E

EBS	Excess Burst Size
ECMP	Equal Cost MultiPath
ECN	Explicit Congestion Notification
ECT	Explicit Congestion Notification Capable Transport
EDR	Event-Dependent Routing
EF	Expedited Forwarding
EGP	Exterior Gateway Protocol
EIR	Equipment Identity Register
EL	Echo Loss
ELR	Edge Label Switching Router
E-LSP	EXP-inferred LSP
EMS	Element Management System
E-NNI	External Network Node Interface
ER	Edge Router
ER-LSP	Explicitly-Routed Label Switched Path
ESP	Encapsulating Security Payload
ETSI	European Telecommunication Standards Institute
EU	European Union
EURESCOM	European Institute for Research and Strategic Studies in Telecommunications

F

FA	Foreign Agent
FEC	Forward Error Correction Forwarding Equivalence Class
FER	Frame Error Ratio
FIB	Forwarding Information Base
FIFO	First In First Out
Fin	Final
FR	Fixed Routing Frame Relay
FRL	Frame Length
FSC	Front-end Speech Clipping
FTP	File Transfer Protocol

G

GERAN	GSM/Edge Radio Access Network
GGP	Gateway-to-Gateway Protocol
GGSN	Gateway GPRS Support Node
GI	General Interest
GMPLS	Generalised MPLS
GMSC	Gateway MSC
GoB	Good or Better
GPRS	General Packet Radio System
GPS	General Processor Sharing Global Positioning System
GRE	Generic Routing Encapsulation
GS	Guaranteed Service (Intserv class)
GSM	Global System for Mobile Communications
GSM-EFR	GSM Enhanced Full Rate
GSM-FR	GSM Full Rate
GSM-HR	GSM Half Rate
GSMP	General Switch Management Protocol
GTP	GPRS Tunnelling Protocol
GUI	Graphical User Interface
GW	Gateway

H

HA	Home Agent
HLR	Home Location Register
HOL	Head Of Line
HSS	Home Subscriber Server
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol

I

I/O	Input/Output
IAB	Internet Architecture Board
ICCC	International Computer Communication Conference
ICMP	Internet Control Message Protocol
ID	Identifier
IDL	Interface Description Language
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IHL	IP Header Length
IIOP	Internet Interoperable Protocol
IKE	Internet Key Exchange
IMP	Interface Message Processor
IMS	IP Multimedia Subsystem
I-NNI	Internal Network Node Interface
IntServ	Integrated Services
IP	Internet Protocol
IPPM	IP Performance Metric
IPSec	IP Secure
IPTO	Information Processing Techniques Office
iptom	IP topology management
IS	Intermediate System
ISDN	Integrated Services Digital Network
IS-IS	Intermediate System to Intermediate System Intra-Domain Routing Protocol
ISP	Internet Service Provider
IST	Creating a user-friendly information society (5th EU research programme)
ITSP	IP Telephony Service Provider
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector

J

JRE	Java Runtime Environment

K

L

LAC	Level 2 tunnel Access Concentrator
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LDP	Label Distribution Protocol
LIB	Label Information Base
LL	Leased Line
L-LSP	Label only inferred LSP
LND	Layer Network Domain
LNS	Level 2 tunnel Network Server
LP	Linear Program
LPC	Linear Predictive Code
LPDP	Local Policy Decision Point
LSA	Link State Advertisement/Announcement
LSP	Label Switched Path
LSR	Label Switching Router
LST	Laplace-Stieltjes Transform

M

MAC	Medium Access Control
MBZ	Must Be Zero
MCU	Multipoint Control Unit
MDRR	Modified Deficit Round Robin
MDT	Mean Down Time
MF	MultiField
MG	Media Gateway
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MGW	Media Gateway
MIB	Management Information Base
MIP	Mixed Integer Program
MO	Managed Object
MOS	Mean Opinion Score
MOSPF	Multicast extensions to OSPF
MPEG	Moving Picture Expert Group
MPLS	Multi-Protocol Label Switching
MRF	Multimedia Resource Function
MRP	Measurement Reference Point
MS	Mobile Station
MSC	Mobile Switching Centre
MSL	Maximum Segment Lifetime
MT	Mobile Terminal
MTU	Maximum Transfer Unit

N

NA	Not Attainable
NAK	Negative Acknowledgement (rejection)
NAP	Network Access Point
NAS	Network Access Server
NAT	Network Address Translation
NCC	Network Control Centre
ND	Network Domain
NDRE	Norwegian Defence Research Establishment
NE	Network Element
NHLFE	Next Hop Label Forwarding Entry
NMC	Network Management Centre
NMF	Network Management Forum
NMS	Network Management System
NNI	Network Node Interface
NO	Network Operator
NP	Network Performance

NPL	Network Parameters Level
NPP	Network Performance Parameters
NRT	Non-Real Time
NTA	Norwegian Telecommunications Administration
NTP	Network Time Protocol
NU	Network User

O

O/E/O	Optic-Electronic-Optic conversion
OA	Ordered Aggregate
OADM	Optical Add-Drop Multiplexer
OAM	Operations and Maintenance
OCC	Optical Connection Controller
OCh	Optical Channel
O-D	Origin-Destination
OIF	Optical Internetworking Forum
OMG	Object Management Group
OMP	Optimised MultiPath
OMS	Optical Multiplex Section
OS	Operations System
OSI	Open System Interconnection
OSPF	Open Shortest Path First
OSS	Operation Support System
OTN	Optical Transport Network
OTS	Optical Transmission Section
overlapPartit	overlapping Partitions
OXC	Optical Cross-Connect

P

P	Provider
PAWS	Protect Against Wrapped Sequences
PBS	Peak Burst Size
PBX	Private Branch Exchange
PC	Personal Computer
PCBR	Premium Constant Bit Rate
PCF	Policy Control Function
PCM	Pulse Code Modulation
P-CSCF	Policy Call State Control Function
PDB	Per Domain Behaviour
PDF	Probability Density Function
PDP	Packet Data Processor Packet Data Protocol (UMTS)
PDR	Peak Data Rate
PE	Provider Edge
PEP	Policy Enforcement Point
PHB	Per Hop Behaviour
PI	Policy Interface
PIB	Policy Information Base
PIM-DM	Protocol-Independent Multicast – Dense Mode
PIM-SM	Protocol-Independent Multicast – Sparse Mode
PIN	Policy-Ignorant Node
PIR	Peak Information Rate
PL	Packet Length
PLC	Packet Loss Concealment
PLMN	Public Land Mobile Network
PMC	Premium Mission Critical
PMM	Premium MultiMedia
PNO	Public Network Operator
POP	Point Of Presence
POS	Packet Over SDH
POTS	Plain-Old Telephony Service
PPP	Point-to-Point Protocol
PRC	Policy Rule Class
PRP	Policy Retrieval Point
PS	Packet Switched
PSC	Per hop behaviour Scheduling Class
PSH	Push
PSPWG	Packet Switching Protocols Working Group
PSTN	Public Switched Telephone Network
PVBR	Premium Variable Bit Rate

Q

QC	Quality Category
QCS	Quality Class Specification
QoS	Quality of Service
QPIM	QoS Policy Information Model

R

R&D	Research and Development
RAB	Radio Access Bearer
RADIUS	Remote Authentication Dial-In User Service
RAM	Random Access Memory
RAN	Radio Access Network
RAP	Resource Allocation working group (IETF)
RAR	Resource Allocation Request
Rec.	Recommendation
RED	Random Early Detection Random Early Discard
REP	Report
REQ	Request
RFC	Request for Comments
RIB	Routing Information Base
RIO	Random Early Detection with In/Out bit
RIP	Routing Information Protocol
RLC	Radio Link Control
RLR	Receive Loudness Rating
RM-ODP	Reference Model for Open Distributed Processing
RNC	Radio Network Controller
ROA	Recognised Operating Agencies
RST	Reset

RSVP	Resource reSerVation Protocol
RT	Real Time
RTCP	RTP Control Protocol
RTFM	Real Time Flow Measurement
RTO	Retransmission Timeout
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
RTT	Round Trip Time
RTTM	Round Trip Time Measurement
RTTVAR	Round Trip Time Variation

S

SA	Security Association
SAP	Session Announcement Protocol
SBM	Subnet Bandwidth Manager
SDH	Synchronous Digital Hierarchy
SDP	Session Description Protocol
SDR	State-Dependent Routing
SDU	Service Data Unit
SG	Study Group
SGM	Small-Group Multicast
SGSN	Serving GPRS Support Node
SIMP	Satellite Interface Message Processor
SIP	Session Initiation Protocol
SIS	Service Instance Specification
SLA	Service Level Agreement
SLND	Server LND
SLO	Service Level Object
SLR	Send Loudness Rating
SLS	Service Level Specification
SM	Simple Multicast
SME	Small and Medium Enterprises
SMTP	Simple Mail Transfer Protocol
SN	Subnetwork
SNMP	Simple Network Management Protocol
SO	SubOptimality
SONET	Synchronous Optical Network
SOS	Service Offer Specification
SP	Service Provider
SPF	Shortest Path First
SPI	Security Parameter Index
SQA	Service Quality Agreement
SRED	Shock-absorber RED
SRI	Stanford Research International
srTCM	single rate Three Colour Marker
SRTT	Smoothed Round Trip Time
SS7	Signalling System no 7
SSL	Secure Socket Layer
SSM	Source Specific Multicast
SSRC	Synchronisation Source identifier
ssthresh	slow start threshold
STM	Synchronous Transfer Mode
STS	Service Template Specification
SYN	Synchronisation

T

T1	Standards Committee T1, Telecommunication (related to Alliance for Telecommunications Industry Solutions)
TCA	Traffic Conditioning Agreement
TCM	Three Colour Marker
TCP	Transmission Control Protocol
TCS	Traffic Conditioning Specification
TD	Tail-Drop
TDR	Time-Dependent Routing
TE	Terminal Equipment Traffic Engineering
TFRC	TCP Friendly Rate Control
TG	Traffic Generator
TIP	Terminal Interface Message Processor
TIPHON	Telecommunications and Internet Protocol Harmonisation Over Networks
TL	Topological Link
TLC	Transport Layer Security
TLV	Type-Length-Value
TMF	TeleManagement Forum
TMSC	Transit MSC
topEndDir	topological End Directionality
TOS	Type Of Service
TRC	Transcoder
trTCM	two rate Three Colour Marker
TTC	Telecommunication Technology Committee
TTL	Time-To-Live
TTR	Time To Repair
Tx	Transmit

U

UA	Unavailability
UAC	User Agent Client
UAS	User Agent Server
UCL	University College London
UDP	User Datagram Protocol
UE	User Equipment
UHO	User Home Organisation
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunication System
UN	User Network
UNI	User Network Interface
URG	Urgent
URI	Universal Resource Identifier
URL	Universal Resource Locator
UTRAN	UMTS Terrestrial Radio Access Network

V

VAD	Voice Activity Detection
VC	Virtual Channel
VCI	Virtual Channel Identifier
VDSL	Very high speed Digital Subscriber Line
VFI	VPN Forwarding Instance
VLL	Virtual Leased Line
VoD	Video on Demand
VoIP	Voice over IP
VP	Virtual Path
VPI	Virtual Path Identifier
VPN	Virtual Private Network

W

WAN	Wide Area Network
W-CDMA	Wideband Code Division Multiple Access
WDM	Wavelength Division Multiplexing
WFQ	Weighted Fair Queueing
WG	Work Group
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin
WWW	World Wide Web

X

xDSL	any Digital Subscriber Line
XFG	Cross Configuration

3GPP	Third Generation Partnership Project