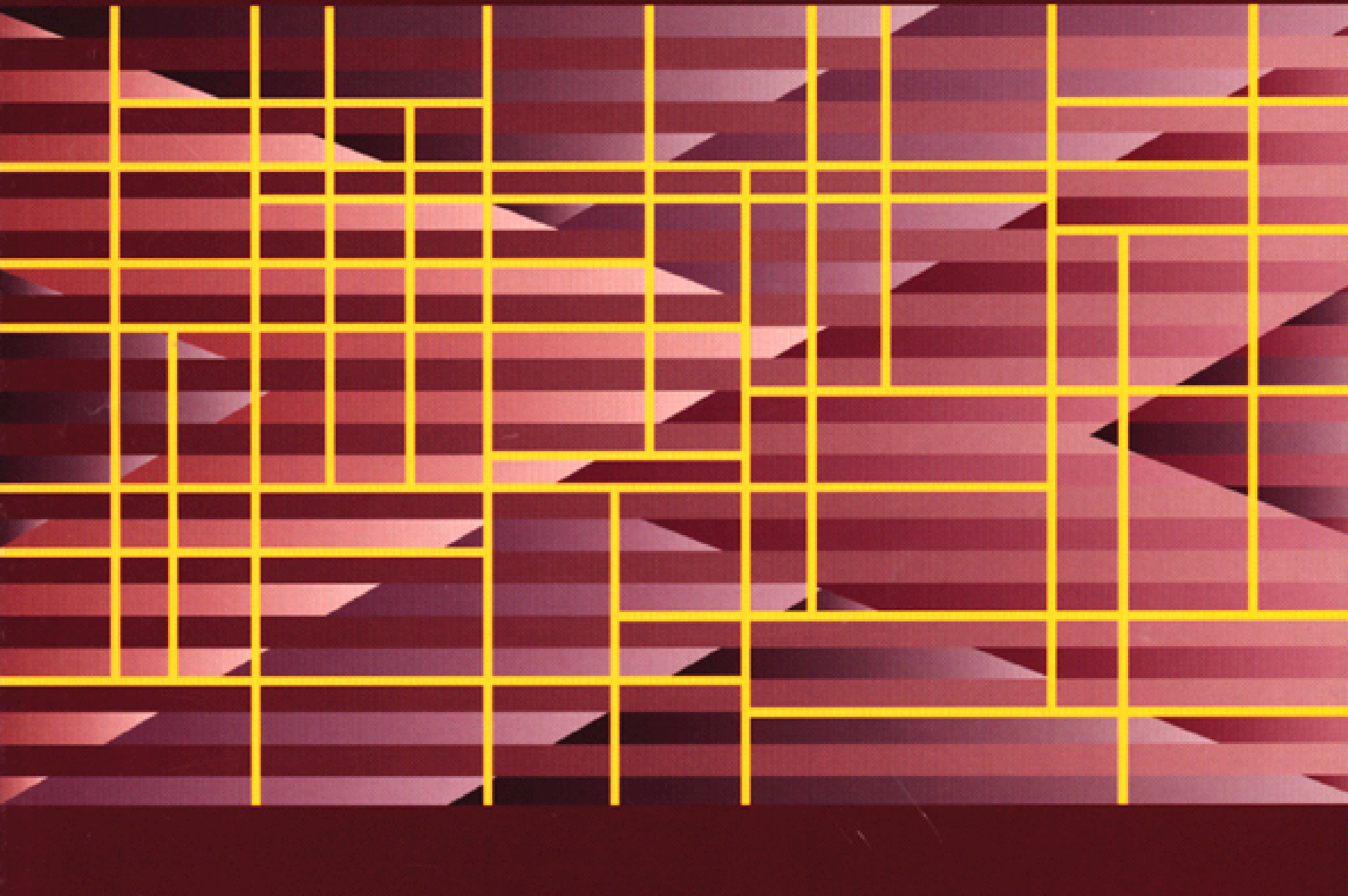
Telektronikk 2.97

Internet



Contents

Feature

Special

Status

Telektronikk

Volume 93 No. 2 - 1997
ISSN 0085-7130

Editor:
Ola Espvik
Tel. + 47 63 84 88 83

Status section editor:
Per Hjalmar Lehne
Tel. + 47 63 84 88 26

Editorial assistant:
Gunhild Luke
Tel. + 47 63 84 86 52

Editorial office:
Telektronikk
Telenor AS, Telenor Research & Development
P.O. Box 83
N-2007 Kjeller, Norway
email: telektronikk@fou.telenor.no

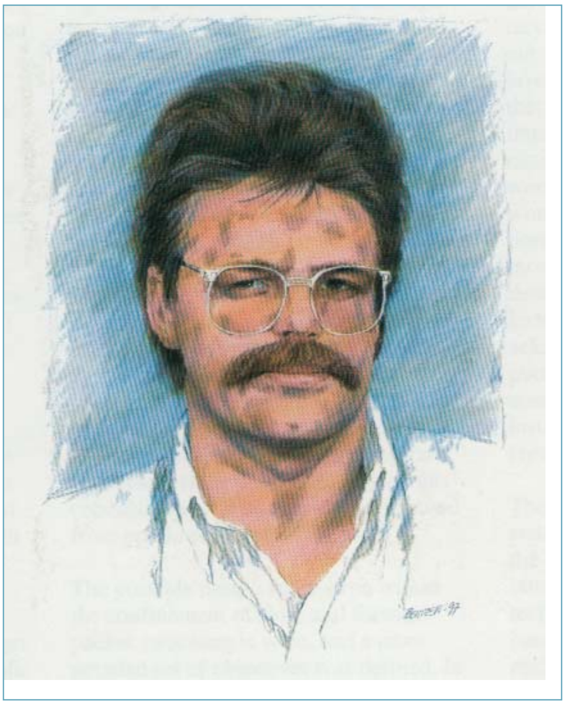
Editorial board:
Ole P Håkonsen, Senior Executive Vice President
Oddvar Hesjedal, Vice President, Research & Development
Bjørn Løken, Director

Graphic design:
Design Consult AS

Layout and illustrations:
Gunhild Luke, Britt Kjus, Åse Aardal
Telenor Research & Development

Guest editorial

ØIVIND KURE



This issue focuses on Internet protocols and architecture. Within Telenor, there has been ongoing research on various aspects of the Internet for the last 14 years. This special issue represents in many aspects the end of a transition epoch. For a fair number of years Telenor R&D was the Internet access in Norway, starting with a satellite link and progressing onto leased lines to England. As the Internet grew, we were pushed further and further out into the branches of the network. Currently, we are only a small group in a multitude of groups within and outside Telenor working on the research and development of Internet protocols and architecture. For a successful technology or architecture, this is the development path. It also represents a challenge to the research environment; in a fast developing field with a multitude of active participants, the research has to be very well focused in order to represent something of value.

The exponential growing number of hosts and network will force a continued development of the architecture itself and the management tools used to administrate the networks. The current problems with the size and management of the routing tables is an indication of the problems that will come. In a future world where perhaps every light bulb could be connected to the network, identifying, authenticating and routing messages to the bulb cannot be handled within the current architecture.

Two of the major challenges facing the Internet is the scaling of the network and the services, and the introduction of guaranteed quality of service into the network. Internet today offers only a best effort service, and the whole philosophy of the network is to offer connectivity. As the congestion in the network grows, more and more users will require some sort of guarantee for their transmission. This is not only to ensure sufficient quality for video and audio transmissions, but also to ensure a reasonable response time for “important” transactions. To guarantee a quality of service will require resource reservation and therefore also differentiated pricing. Usage based pricing is to a large extent different from the current scheme, where the bit transport in the IP network seldom is metered. In addition to all the technical challenges of pricing, metering, and resource reservation, there is also a question of user acceptance which is the ultimate essence of quality of service.

The solutions to these problems are necessary for a continued growth of the Internet, but they are not sufficient. New user requirements will have to be met. Ted Lewis points out in his latest column in IEEE Internet Computing (vol 1, No 2), that most development within computing has been driven by metaphors. The Internet started out as a sort of printing press for rapid distribution of printed words. Currently, the possible transition towards Internet as a phone and Internet as a TV has started. Whether these will succeed as the metaphor driving the development of the Internet is unknown.

Personally, the metaphor I believe will drive the development is Internet as a universal dial tone; the next generations of the Internet should be something that is available wherever you want it, reaching whatever you want.

Øivind Kure.

Basic Internet Protocol (IP)

ØIVIND KURE

The intention of this article is to provide a sense of the functionality and limitation of the protocol suite that imprecisely is called Internet or TCP/IP. Internet is a global network of networks running the IP protocol as the network protocol.

The TCP/IP protocol is used as a sloppy term for a whole suite of protocols. More than 2000 documents describe past or current versions of the protocols in this suite. Although TCP (Transmission Control Protocol) and IP (Internet Protocol) are stable protocols, new protocols are continuously being added.

The emphasis is on the philosophy behind the design and most of the focus is on the network layer protocol. This is the foundation of the protocol suite, and it is the element that is the most difficult to change. Any change to the network layer requires that all machines and routers within the network must adopt the change. This is in contrast to changes to higher layer protocols. All other application and transport services are built by adding functions in the end system protocols. Adding a new protocol only requires that the endsystems interested in the protocol implements it. A new protocol can therefore gradually be added into the network. New protocols can and will therefore continuously be added as the application requires new and different services. The network layer is therefore the component with the highest cost of changing, and understanding the IP protocol therefore provides the key to understanding the Internet.

In order to do this, it is necessary to understand the philosophy behind the design. The Advanced Research Projects Agency (Arpa) of the US Department of Defense started the development of the Internet protocol suite in 1973. The protocols, in particular the TCP and IP protocols, were designed based on 8 goals that had to be achieved. The best description of the design philosophy can be found in the articles by J. McQuillan et al. [2] and D. Clark [3]. Some of their work is summarized in the article.

One of the premier reasons for the success of the Internet protocol suite is the suitability for interconnection of heterogeneous networks; the cost of starting to use the protocol is lower, since existing network infrastructure can be reused. Indeed, the overall design goal was exactly this; to develop an effective technique for multiplexed utilization of existing interconnected networks. Although the goal was envisaged before Local Area Networks (LANs) emerged, the principle used for interconnection was flexible enough to be suitable for these technologies too. Originally, the protocol suite was intended for interconnection of the original Arpanet and the Arpa packet radionet, and possible future networks that would evolve [3].

The Internet network layer is based on store and forward packet switching; packets are received and buffered until they can be sent forward. Packet switching was chosen since the networks to be interconnected were packet based. As a natural consequence, store and forward packet switching was chosen, since this concept was well known and understood from previous projects.

The possible design space even within the confinement of store and forward packet switching is wide, and a more detailed set of objectives was defined. In prioritized order they were:

1. Internet communication must continue despite losses in networks or gateways.
2. The Internet must support multiple types of communication services.
3. The Internet architecture must accommodate a variety of networks.
4. The Internet architecture must permit distributed management of its resources.
5. The Internet architecture must be cost effective.
6. The Internet architecture must permit host attachment with a low level of effort.
7. The resources used in the Internet architecture must be accountable.

This is a collection of design objectives that nearly all internetwork technologies could have used. As we will see through the discussion, the unique element is the ordering. Only the most important objectives are achieved, while some of the less important ones are fulfilled with varying success. Simplicity in the network was traded for complexity in the hosts. Interconnection of networks was traded against the more efficient unified network for multiple media.

Survivability was ranked as the most important issue. It was interpreted severely and restated to: an error in the network would constitute only a temporary disruption of the communication and not require a complete reset of higher layers communication. The only error that should be reported to the user of a transport layer was the complete partition of the network. This implied that the network had to be stateless, so no error would disrupt the state of a communication. In communication between two or more hosts there has to be some state describing the status of the conversation. Examples are number of packets acknowledged, number of packets transmitted and number of outstanding flow control packets. If this information is lost, data can be lost and thereby the synchronization between the applications.

The two options are either keeping the state at the end-points, the hosts, or store the data resilient in the network. The latter would require using secure storage techniques similar to those used in database transactions. By keeping state at the endpoints, only errors in the hosts could affect the state and the communication; if the host failed, the communication would also fail. Clark calls this the "fate sharing" approach [3].

It is not the most efficient design, since higher layer protocol cannot assume that error will be reported by the network. The network may make an effort, but any assumption about ability to detect errors must be based on probing by the hosts. Such probing also has a limited granularity; the details of the errors detected will be more limited compared to those the network potentially could have reported.

Survivability also meant that the routing within the network should be dynamic. Each packet should find its own way through the network. Within such a framework connection admission control does not fit in. The focus of the network is to connect endpoints with varying paths; not to give any guarantees on the quality of the path. The adding of services with specific qualities of service therefore requires a change in the network layer.

Support of many different services was ranked lower in importance. The services could range from a reliable byte transfer to real time transfer of voice. The focus of the protocol suite was for the two transport services provided by TCP and UDP (User Datagram Protocol), the first provide a reliable byte stream transfer,

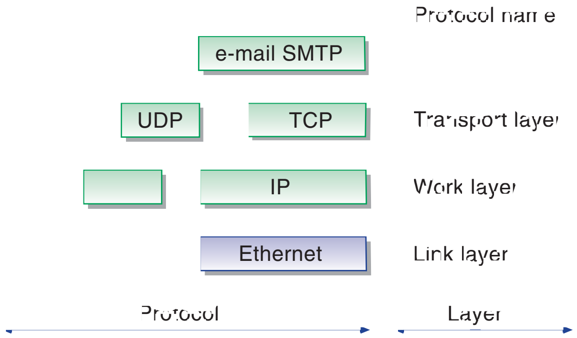


Figure 1 The protocol model for the Internet protocol stack

the latter an unreliable packet transfer. The important impact of this goal was the separation of the network and transport layer. Originally, they were designed as one layer. However, the requirement of multiple services forced a separation. The network layer, IP, provided an unreliable datagram service. On top of this, TCP used acknowledgment schemes to provide a reliable transfer.

The next design goal was the ability to incorporate a wide variety of technologies. This has been one of the most important success factors of the Internet; the cost of utilizing existing infrastructure is low. The Internet is running over anything from Gigabit Ethernet to low bandwidth radio packet networks in the 100 baud per second range. This can only be achieved by making almost no assumption about the underlying network technology. IP assumes only that the technology will make a "best effort" to deliver a datagram of a reasonable size. Clearly, the packets have to be somewhat larger than the header for the network to function. The network layer is based on the absolute lowest denominator of functions. No assumptions are made on delay, delay jitter, order of delivery, or any other network service like multicast or priority transmission. This is a direct consequence of the objective to achieve efficient interconnection of networks. Any other assumption of the functions of a network technology would have required refitting functions into existing networks or building separate network service enhancement gateways. The latter would also increase the vulnerability of the network.

Any enhancement of the unreliable datagram transfer was instead added in the end system. This implied added complexity in the end systems. Although low complexity of the host was an objective, it was ranked lower than support for survivability, multiple services, and network technologies. The implication was therefore, given the choices made on network architecture, to use an architecture with the lowest possible complexity.

The ordered list of objectives resulted in a network based on unreliable datagram transfer, where any additional service was implemented in the end systems. Clearly, such an architecture may use network resources inefficiently; regardless of where a packet is lost or damaged, it will have to traverse the whole network again.

It also poses a challenge for the control of the network. Flow and congestion control had to be based on end-to-end mechanisms with little feedback from the network itself. Assuming well behaved hosts, mechanisms can be found. However, they are vulnerable for hosts choosing to operate contrary to the control mechanisms; the network has no policing mechanisms and there is no contract on how a communication stream should behave. All is geared towards everybody behaving according to a global optimum, getting a fair share of available bandwidth.

Intentional "misbehaving" hosts has not been a problem. Instead, the problem has been the shift from traffic over the reliable transport protocol TCP with flow and congestion control towards the unreliable transport protocol UDP which lacks these mechanisms. The latter will by virtue of not having a flow and congestion control always get an "unfair" share of the bandwidth compared to a TCP connection.

The list of design objectives can also be viewed as a summary of where the challenges for the Internet protocol suite will lie in the future. Foremost is the delivery of any quality of service. This is contrary to the philosophy, namely assume hardly anything about the underlying network technology. Delivering a transfer service with given quality of service forces resource reservation. The moment resources are reserved, there has to be some sort of charging and accounting; otherwise every communication stream would receive the highest priority and the same quality of service.

Although resource reservation and Quality of Service (QoS) is outside the scope of the current Internet it can be implemented to various extents. However, it can only be done through making further assumption about the underlying technology. Islands of networks can be designed, where Quality of Service for IP is offered as long as the communication is within the island. The current work on multiprotocol label switching falls under this heading. Although it is not aimed at providing QoS, it is a proposal for a tag switching that among other things could be used to implement QoS. ATM (Asynchronous Transfer Mode) can be used in a similar fashion. Each individual connection between applications are mapped onto a separate ATM channel with it own Quality of Service. Within such an island IP is used only as a framing protocol, but once the connection is outside it reverts back to its original role as a network layer. This is discussed further in the article on the next generation of IP protocols, Ipv6.

Framework

The most important framework of the Internet is the belief in open standards. The protocols need not be defined by a company or organizations before they are implemented. The spirit of the process is as follows; with the IP protocol as a common layer, users that have requirements for application/middleware or protocols are free to define and code them. The ones that fill a need will be adopted by the community and sometimes rapidly

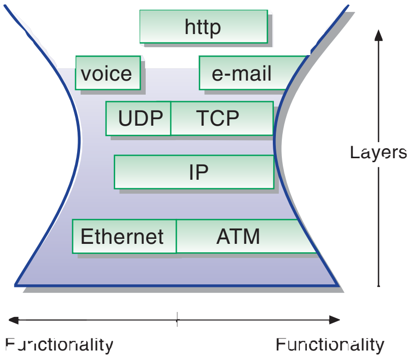


Figure 2 Displays the functional relationship between the layers. New layers add functionality based on a subset of the functionality of the underlying protocols

become a standard. The others will fade away. This dynamic process is the core of the success of the protocol suite. The protocols that become standards are open and have been thoroughly tested in the marketplace both in terms of functions, availability and administration.

The framework consists of two elements: the model of the protocol stack, and the addressing. The former addresses the relationships the various protocols have to each other, the latter describes how entities are identified within the structure.

Protocol architecture model

The Internet protocol suite consists of four layers; an application layer on top, followed by a transport, network and data link level. The data link level contains the protocols necessary to utilize a particular communication link and network interface card. The network layer handles the forwarding of packets. The transport layer builds an end-to-end service, that is offered to the application layer. The model does not have separate layers for management and control. Instead these functions are incorporated into each layer, often as separate protocols. For example, the network layer contains routing protocols, the IP protocol, and the ICMP (the Internet Control Message Protocol – see separate box).

The hour glass model in Figure 2 shows the relationship between the various protocols. It also provides a sense of the architecture. The horizontal axis is a measure of the functionality, where the width is a measure of the set of functions. Clearly the axes are more of a qualitative measure. The IP protocol is the core providing the building block all applications rely on. Protocols at higher layers utilize a subset of the functionality of the underlying protocol and adds more or “better” functionality. IP is based on the lowest common denominator and therefore only utilizes a subset of the functionality of the underlying protocols. The figure is included to demonstrate how the IP protocol is the core of the protocol stack. As more functionality is required, new protocols building on all or a subset of lower layer functionality are added.

Figure 3 shows the relationship between the various layers for a WEB client communicating with a WEB server in a different administrative domain. The IP protocol is the glue interconnecting the different networks and technologies. Since different transmission mediums are used, the link protocols also differ. For dial-up clients the Point-to-Point protocol is the most popular one. In the example in the figure, ATM is used as the link level technology since this is gaining a wider acceptance as the technology of choice for backbones.

Internet Control Message Protocol (ICMP)

This is the protocol for exchange of error and other control in messages in an IP network. It is carried with an IP datagram, but it is a separate protocol. There are 15 types of error and control messages, each type with one or more subtypes (called codes). Among the more important ones is the type signaling a destination is unreachable, a redirect type, source quench, time exceeded, and echo request and reply. The Internet therefore has the capability of a limited feedback of error information from the network.

Addressing

There are four levels of addressing; logical addressing of machines based on administrative domains, a transport address, a hierarchical network address, and a link level address. In addition, there can be protocol specific addressing schemes used, like The Universal Resource Locator in WEB.

To exemplify the addressing, my old lab machine has the logical name *brage.nta.no*; identifying it as administratively being in Norway, within the organization identified by *nta* (Nor-

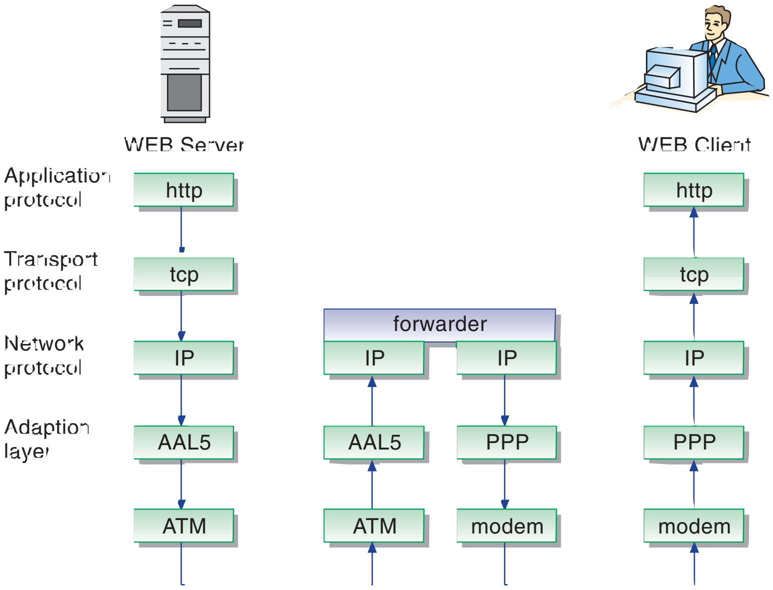


Figure 3 The path through the protocols for a message from a WEB server to a client

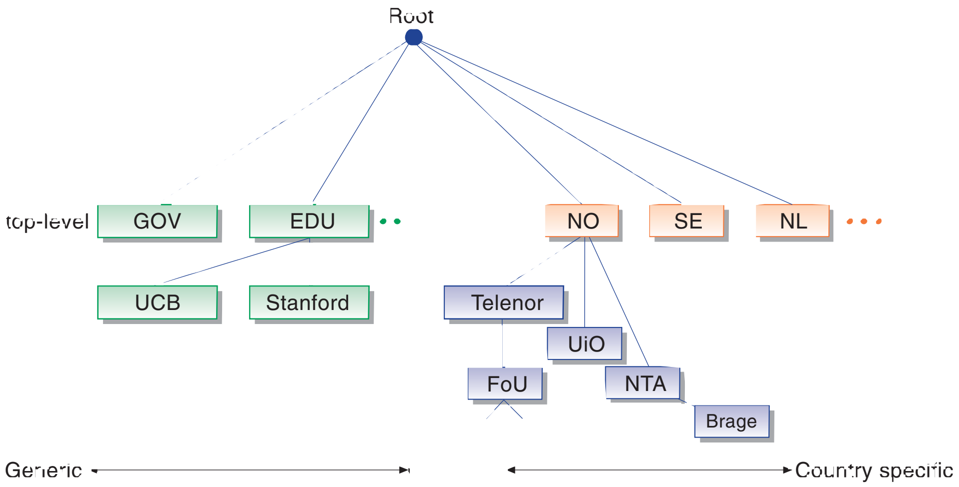


Figure 4 Organization of the naming space

wegian Telecom Administration) and with the name *Brage*. The IP address of the network connection is 128.39.20.3. The link level address of the Ethernet interface card is 8:0:20:c:78:ff. In order to identify the application protocol, a transport level address called port is used.

Logical addressing

The logical addressing provides names with a structure as a form of addressing a machine or host. It is called the Domain Name System (DNS). It is a hierarchical tree structure. The name of a machine will be written in the form *brage.nta.no*. The suffix *(no)* is called the top level domain, while the last dot is called root, the entity owning the whole structure. As will be explained later, the authority of the DNS structure is distributed so the root now serves no function.

The way it is organized can best be explained by starting at the requirements. The purpose is to be able to use logical names that can be mapped to an IP address [4]. In order for humans to be able to navigate in the name space, it needs a hierarchical structure with several levels. With a network of networks like Internet that has an exponential growth rate, the mapping from name to address had to be done as a distributed application [4]. The scale of the problem prohibited any central authority that would be responsible for updating and maintaining the mapping. The distribution of the updates cannot be pushed to all connected hosts, otherwise the updating traffic would be prohibitive. The solution is a distributed database with a tree structure, where updates to the mapping are distributed only on requests.

Top level domains are grouped in specific and generic domains. Examples of the generic ones are *com* for commercial organization, *gov* for US government, *net* for network operators, and *edu* for universities. The specific domains are organized by country, with *no* for Norway, *se* for Sweden, and so on. Under each of these top level domains other domain suffixes can be added to identify organization or other classification structures. The system is open ended, since new top level domains can be added. There is a continuos discussion of adding more top level domains, due to the proliferation of systems being added.

In the mapping database, only trusted individuals are allowed to update the information. Otherwise, even without malignant users, the misconfiguration alone would have killed the system. The delegation and distribution of trust is therefore an important issue. Again for the system to scale, the number of individuals that are trusted to update the system has to grow roughly in the same order as the number of hosts. The structure of the database reflects the delegation of authority to change and update. An entity (organization or company) is responsible for each of the top levels. It grants authority to other entities for one or more of the domains within. These are called zones. Each zone can be subdivided into smaller zones by adding additional levels in the naming structure. As zones are split into new zones, the authority of the zone is also transferred. The entity or person responsible for a zone is responsible for having all data on that particular zone updated. For a zone subdivided into other zones, this is reduced to containing the IP address of the

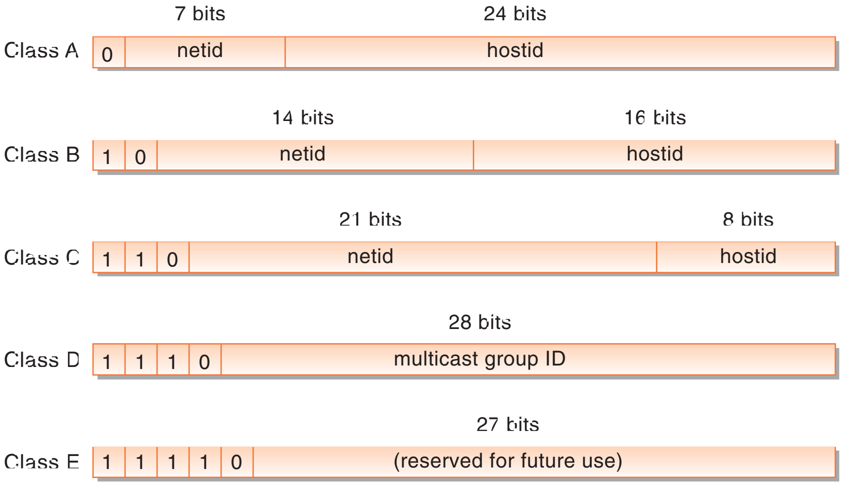


Figure 5 The structure of the IP address

Table 1

Class	Range	
	From	To
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

machine with the information of the subzones.

Through this architecture, the name space will scale. As new branches are added in the tree, only the one directly above the new branch must be aware of the change. In order to find the IP addresses corresponding to a logical name, one needs to find a node responsible for one of the zones in the name. From there on it is just to follow the information given in the corresponding zones.

The final element that enables scaling, is caching of information. All mapping between name and address has a lifetime. For that period, usually in the order of hours, the mapping will be valid. All servers in the network will cache the results of mapping request. If the information in the cache is still valid, the request will be answered from the information in the cache.

Network layer address (IP address)

In the Internet, each interface has a unique network address. A host with several interfaces will therefore have several IP addresses. The addresses are 32 bit long, divided into a network id and a host id part. Figure 5 shows the structure of the address. The address is normally written in a dotted-decimal notation as 4 groups of numbers. The ranges for the various classes of addresses are shown in Table 1. The addressing structure is inefficient, since few networks have enough hosts connected to fully utilize the host addresses space. The host id part is therefore split into a subnet part and a remaining host id part. The number of bits allocated to the subnet id is up to the local administrator, since it has significance only within an administrative domain.

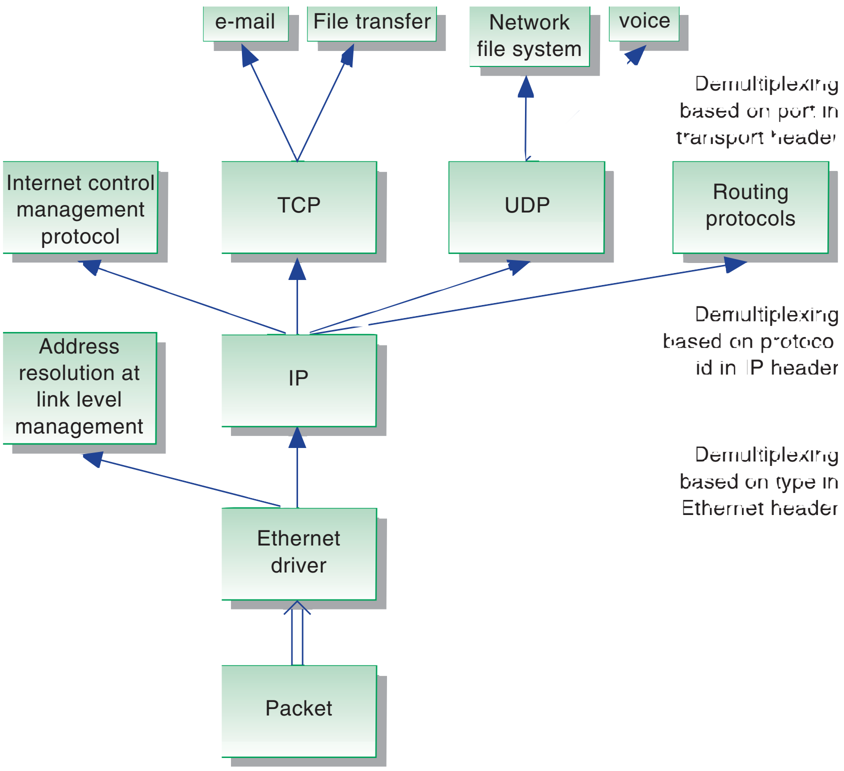

e-mail
File transfer
Network file system
voice
Demultiplexing based on port in transport header
Internet control management protocol
TCP
UDP
Routing protocols
Demultiplexing based on protoco id in IP header
Address resolution at link level management
IP
Demultiplexing based on type in Ethernet header
Ethernet driver
Packet

Figure 6 The demultiplexing hierarchy

Protocol stacks

Layering

The TCP/IP protocol stack is a combination of protocols at different layers. The following section will describe the functional interface between the layers, the multiplexing and demultiplexing of the protocols.

The demultiplexing is conceptually the simplest, since only limited catalog services are required. The demultiplexing consists of one common function, the

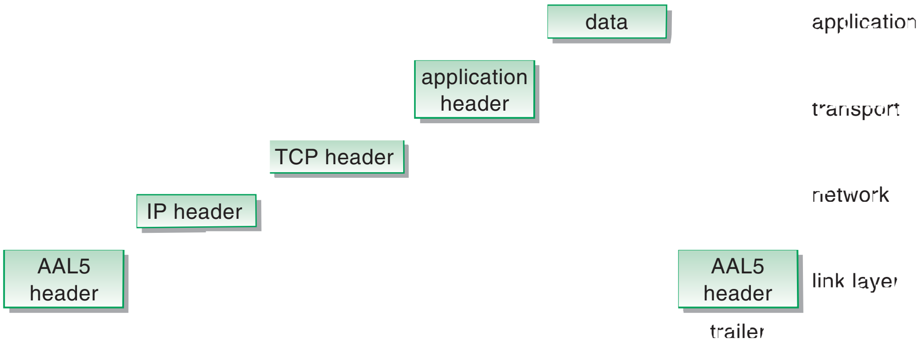


Figure 7 The non-standard encapsulation hierarchy with IP over ATM

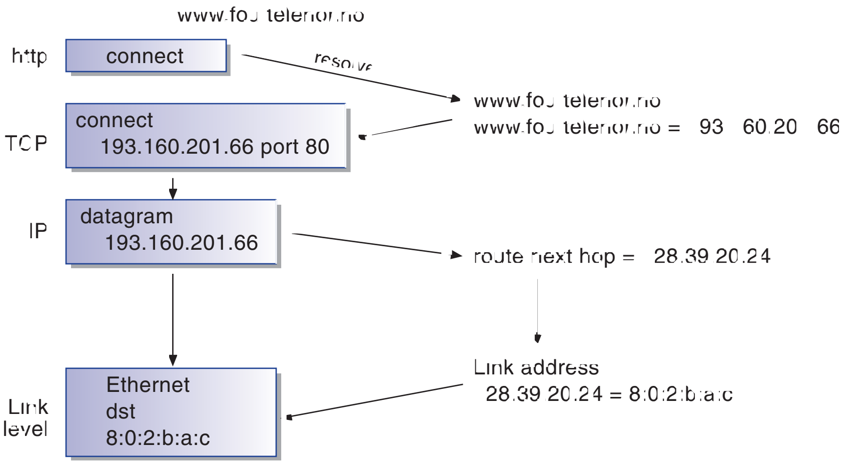


Figure 8 The hierarchy of address resolution when sending a message to Telenor FoU's WEB server from the author's machine

recognition of the address of the protocol in the next higher layer. In addition, each layer will have specialized functions. The demultiplexing is illustrated in Figure 6. The largest number of mappings exist at the transport level. The Internet uses the concept of a port number to designate the application, and there is a range of port addresses for well known application protocols. The mapping between port address and application is stored in a local file or catalog service.

The encapsulation is conceptually more complex since several catalog services are required. The encapsulation is straight forward and illustrated in Figure 7. Each layer adds its own header at the beginning of the datagram. Although straight forward, it is slightly inefficient; checksums and integrity fields are calculated over the whole datagram. Adding these fields as trailers, at the end, would have been more efficient.

Figure 8 shows the number of catalog services that are used during the encapsulation. Several are required at the initiation of a connection between two applications. Normally, the address of the host of the remote application is given as a logical name that must be mapped to an IP address. This is done through the name database.

The next step is to identify the next hop the datagram should be sent to. At each system, both end system and gateway, there is a database containing a mapping between destination address and address of next hop. For many end systems, the database contains only one entry, the address of the gateway handling all forwarding. Most gateways will only have a limited knowledge of their environment. For all other destinations they will have a default hop for all destinations that are unknown. In this fashion, the packet will make its way deeper into the network.

The next hop is not carried as a field in the IP protocol header. There has to be a direct connection to it, so the IP address is not needed directly. However, on a local area network several host interfaces can be connected, so the link level address is needed. The mapping between IP address and link level address is technology specific, since the addressing scheme depends on the technology. For legacy networks (Ethernet, token ring and FDDI), the Address Resolution Protocol (ARP) handles the mapping between IP address and link level address. For ATM networks there are similar protocols with the same functionality and Application Programmers Interface (API).

Fundamentals of the IP protocol

Protocol functions
Best effort transfer

IP is a network layer that offers an unreliable, connectionless datagram service. The are no guarantees that a datagram will be delivered to its destination. The routers, constituting the IP forwarding mechanism, will make a best effort to deliver; should there be any problems, like lack of buffers, the datagram will be discarded. If a reliable service is required, it will have to be provided by one of the upper layers on an end-to-end basis.

Contrary to ATM, there is no guarantee on the order datagrams are delivered in. IP handles each datagram independently. Each router has a forwarding table giving the link and address of the next router to a particular destination. These paths are updated by separate routing protocols. The routing information is updated independent of the packet streams, so two sequential packets may therefore take different paths.

IP is designed to be the network layer on top of heterogeneous technologies. The protocol is therefore based on very few assumptions of the properties of the underlying technologies. In principle, the path a datagram will take is not determined at the beginning of a transmission; the size of the data can therefore not be adapted to the maximum frame length of the underlying technologies that will be traversed. The network layer must therefore have the ability to fragment a datagram into elements that will fit onto a particular technology. For the same reason, the routers do not have the ability to reassemble fragments; all fragments may not take the same path, and it is in addition a costly process to implement in routers. Instead, reassembly of fragments is done in the end system. The only assumption IP makes is that end systems are at least able to handle 576 byte long datagrams.

Beyond these basic functions, the protocol also allows for some optional functions. The optional functions are rarely used, and they deal mainly with how the forwarding is performed.

Protocol header

The protocol header is displayed in Figure 9. It has a minimum size of 20 bytes. The fields reflect the services offered by the protocol. The connectionless service implies that each datagram must contain the source and destination address; the forwarding is done solely based on the fields contained in the datagram. The information must therefore be protected by a header checksum to detect bit errors during transmission and handling in the routers.

The forwarding of packets is a best effort. The routers will forward based on a belief in what constitutes the best path to a destination. Such a path may contain loops due to changes in the path while the datagram is forwarded. To guard against a network filled with packets looping around, each datagram has a time to live field. This constitutes the maximum number of routers a datagram can be forwarded through, and at each forwarding the number is decremented. By guaranteeing that packets will have a limited life, the identifier fields in higher layer protocols can be shorter, since they can be reused after a grace period.

The ability to fragment datagrams during the forwarding implies that each fragment must contain sufficient information to be reassembled: the packet the fragment belongs to and where in the packet the fragment fits in. In the IP packets the fields to carry this information is the packet identifier, a unique number per end system, the total packet length, and the fragment offset. In addition, there is a flag field to signal beginning and end of fragment and to set the option that a datagram should be thrown away instead of being fragmented.

In each datagram there is a protocol field used to designate the upper layer protocol the content is handed over to. Since IP is connectionless, the identification of

PPP

The Point to Point protocol is the most frequently used protocol used for dial up connections to the Internet. It is used over a modem line between a PC and the access server controlling the modem pool of an Internet access provider.

The protocol has three components; the encapsulation of an IP datagram on a serial line, a link control protocol to establish and configure the link, and a set of network control protocols. The encapsulation and link control functions are straight forward and contain functions for 1) error detection, 2) frame encapsulation, and 3) protocol multiplexing. The PPP protocol can be used as link protocol for serial lines also for protocols other than IP.

Currently, network control protocols are defined among others for IP, Decnet, OSI CLNP. For IP the network control protocol contains functions for compression.

The datagram in a TCP stream, the TCP and IP headers are each 20 bytes. For applications with short messages, the length of the headers constitutes a substantial delay for a slow 28 kbit/sec modem line. However, over a serial line, few of the protocol fields are likely to change in an unpredictable way. By precalculating state at each end of a serial line, the protocol headers can be replaced by 3 to 5 bytes of state information. Normally, up to 16 different connections can be supported. Connections outside this range are carried uncompressed.

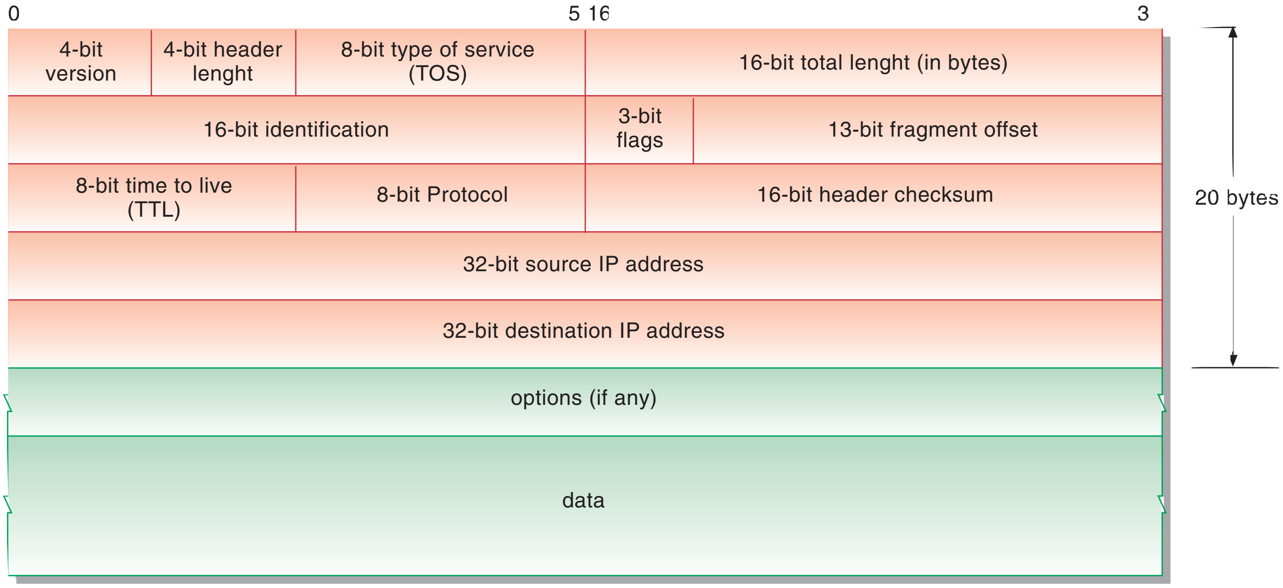


Figure 9 IP protocol header

the transport protocol has to be carried in each datagram.

The remaining fields are related to protocol administration and efficiency. There is a file identifying the protocol version. The current version, IP version 4, will eventually be replaced by the next version, No. 6 or Ipv6. For efficiency, there is a header length field included. The options make the IP headers of variable length; the parsing of the protocol header is made simpler by giving the total length of the header as one field in the header.

The only field in the header we have not discussed is the type of service. It was intended as a way of indicating what kind of application that had originated the data content. Since applications have different requirements on delay and throughput, this field could have been used to provide some sort of priority mechanism in the forwarding. However, the mechanism was never used by the various implementations, and all packets received the same service by the routers.

The optional functions are represented by optional fields in the header. The options always end at a 32 bit boundary, ensuring that the IP header is always a multiple of 32 bit.

Limitations and the path forward

As I have tried to describe, the IP protocol was designed with a limited scope. It therefore has several inherent limitations. These limitations are being addressed in the next version of the IP protocol, Ipv6; the description of the limitations is therefore also a map to the future.

The one limitation that may seem most apparent to the public is the address space crunch. Although the address space is 32 bit long, it is inefficiently used. A lack of available addresses is therefore apparent. As an example, only class C addresses are available now, but these can contain at most 254 hosts. Even a medium sized organization is therefore required to have several class C networks.

The lack of address space is one of the initiators behind developing a new generation of Internet protocol, the IPv6. The use of subnets has been the temporal fix. By setting aside a variable subset of bits of the host part of the IP address, the address space could be better utilized. However, this has implications for the routing and scaling of the network. If the subnetting information were to be distributed outside an administrative domain, the routing tables would also grow substantially. The Internet has used the opposite approach, within the backbone, groups of class C addresses are treated as one routing unit.

The logical addressing also has inherent limitations due to the lack of structure of the type name, street, town, zip code, country type. There are top levels based on either country or on type of organization. The latter level is proposed being expanded to also include levels based on types of business. It adds structure, but makes search for a partial address harder. However, the structure of the logical names is as much a problem due to the uniqueness of the network. On the Internet the national boundaries will to some extent disappear. A logical addressing based on geography will therefore in some cases be meaningless. These are uncharted territories that still have to be explored.

The principle of assuming as little as possible about the underlying networks, implies an inefficiency in the operation of the network. As already mentioned, since reliability is an end-to-end issue, datagrams that fail somewhere in the network will have to be retransmitted and traverse the whole network again. Another source was the lack of report on the capabilities of the underlying networks. The longer packets an application can send its data in, the more efficient is the network.

The Maximum Transfer Unit (MTU) of the underlying technologies varies, from 256 bytes in the old packet radio networks to 64 kbytes in cluster and memory interconnect technologies. To avoid fragmentation in routers, packets for a destination outside the network is typically limited to 512 bytes. However, ICMP protocol can be used to discover the largest possible MTU on a path. This is done by sending a datagram with a no-fragmentation flag set. Routers that cannot forward the datagram without fragmentation will drop the datagram and return an error message. The largest possible MTU can be found by increasing/decreasing the datagram size, but at the cost of causing errors in the network. Although this detailed description is outside the scope of the article, it is included as an example of the flexibility of IP and the lack of effective management tools.

With a design objective of making minimal assumptions of the underlying network technology, security has to be an end-to-end issue. The network does not provide any guarantees on confidentiality, integrity, or authentication. The former two always have to do on an end-to-end basis in order to secure the communication all the way to the application. While this is always the case, having no support in the network itself for these functions, may be inefficient and more vulnerable to mismanagement. Secure versions of a network layer have been developed. In the protocol SP3, a gateway will encrypt the datagrams on a connection and encapsulate them in a regular datagram addressed to the gateway that can unpack and decrypt them. The destination gateway can then send the datagram to the original destination. In this fashion, traffic from secure domains can be transmitted securely over an insecure network. Any support for these in the network itself is only a question of effective performance. The lack of these within IP is therefore not a functional limitation. Authentication of network elements is on the other hand a limitation.

Previously, the lack of a guarantee of the QoS on a connection has been identified as a limitation. This was clearly outside the scope of the design objectives. However, for network islands based on technologies where QoS can be guaranteed it is possible to let the QoS of the underlying technology also encompass the IP layer. For example, there are no technical difficulties in mapping each transport connection to a separate ATM channel. Ignoring the effect of protocol overhead, IP over ATM can have the same functional capabilities for offering QoS as ATM itself.

Multicast, i.e. the ability to address multiple hosts on one connection has been added to the Internet by adding new protocols. IP was designed for point-to-point connections and broadcast connections. The selection of multiple destinations for one packet was outside the scope of the forwarding and routing mechanisms. It has been added later by overlaying the IP network with a new network that has multicast capabilities. This overlay network, Mbone, uses IP as a network layer, but has different routing protocols, forward mechanisms, management, and transport services. The multicast networks are used for limited broadcasting of conferences, events, and distributed simulations. It is an area of active research, since the scaleability of large multicast networks is a fundamental problem.

Conclusion

The main idea has been to convey the basic ideas behind IP. IP is the protocol that sets the limitations and potential for the Internet. It has been an extremely successful and well designed protocol. However, it stands before two major challenges, the continued growth of the network, and the switch in paradigm, from best effort to quality of service. QoS will have to be included to some extent. It will require resource reservation. Resource reservation also implies differentiated cost and pricing and blocking, i.e. some may not get access due to the network being busy. All of these issues are familiar in the existing phone network. These latter points are contrary to the philosophy behind the Internet, and their introduction is not only a technological problem but also an organizational one for the existing Internet.

Bibliography

1 Stevens, W R. *TCP/IP Illustrated, Volume 1.* Addison-Wesley, 1994.

2 McQuillan, J, Walden, J. *The ARPA Network Design Decisions Computer Networks,* 1, 1977, 243–289.

3 Clark, D. The Design Philosophy of the Darpa Internet Protocols. *Proceedings of SIGCOMM 88,* 1997, 106–114.

4 Mochapetris, P, Dunlap, K J. Development of the Domain Name System. *Proceedings SIGCOMM 88,* 1988, 123–133.

Øivind Kure is Senior Research Scientist at Telenor Research & Development, Kjeller. His main research interests are performance analysis, high performance servers, and networks architecture.

email address:
oivind.kure@fou.telenor.no

The World Wide Web Revolution

JON S. VON TETZCHNER

The World Wide Web and Internet have become a key component in computer usage during the last few years. The Web technology, invented in 1989, has some important elements that have made it popular, the most important one being simplicity, both in creating documents and in use.

The most important part of the Web is the protocols HTTP and MIME and the HTML document format. Additions to the Web, such as JAVA and JavaScript are changing the way the Web works. How these new additions are used may decide whether the Web will continue to be a great tool for all users or whether these new additions will change the Web into a service for the majority, while minority groups, such as people with handicaps, will be left on the sideline.

A little history

It is unlikely that Tim Berners Lee knew what he was starting as he worked on making the World Wide Web. When in 1989 he proposed the protocols that make up the Web, it was meant to be used as a means to distribute information among energy physics research groups [Savola 1995]. Now the Web is becoming a great commercial service for information and trade.

Although the Web technology was created in 1989 it was not until 1993, with the creation of Mosaic, the Web browser, that the Web really started to become popular. In 1993 there were only a few information servers around, while a year later there were tens of thousands. Now there are more than a million.

Mosaic made the Web easy to access. By clicking links in the documents, users could surf between documents from all corners of the world. Soon there were alternative browsers on the market, but Mosaic really was the first big success.

The Web technology

The Web is based on a client/server technology. The server contains the documents that the client (browser) shows. The documents can be of any format, but for navigation, the documents should be of the Hyper Text Markup Language (HTML) format. HTML is part of the Web technological base.

Documents are fetched from the server using the Hyper Text Transfer Protocol (HTTP). Both HTTP and HTML are described in more detail later in this document.

There are 4 main reasons for the success of the Web:

1. It is very easy to make documents in the HTML format. Any user with a little computer knowledge can learn to make simple documents in a few hours or days.
2. HTML documents can be displayed on just about any screen or device.
3. It is very easy to browse between the documents. Non-technical users learn the basics in hours.
4. Many Web browsers allow access to FTP, Gopher and Wais documents as well. This means that the same tool can be used to access a vast amount of information and that investment in other information systems was not in vain.

MIME is another key component in the Web technology. MIME is a way to tell the browser what kind of information, or in what format, it is receiving. Typical MIME types are:

text/plain	Text file
text/html	HTML document
image/gif	GIF graphics file. The GIF graphics format is the most used graphics format on the Web at the moment.
image/jpeg	JPEG graphics file. JPEG is the second most used graphics format.

MIME types have been defined for a lot of different graphics format, audio, video and text documents.

The HTTP protocol

> *An HTTP session is short and sweet. The client opens a connection to the server, sends a request, and awaits a response. When the server receives the request, it generates a response, sends it to the client, and closes the connection.*
>
> Dave Roberts

The HTTP protocol is used for communication between the client and the server. The client requests information from the server using HTTP requests. When the request has been answered through data or some kind of error message, the server disconnects the connection (HTTP 1.0).

A standard HTTP request has the following structure:

```
<method> <path> <protocol version>
<request header fields>
```

A typical request might look like this:

```
GET /download.htm HTTP/1.0
User-agent: Opera 2.1 (Windows 95)
Referrer: http://opera.nta.no/opera.htm
```

The only required statement is the first, which requests the document 'download.htm' from the HTTP Web server. The protocol version to be used is HTTP 1.0. The second statement tells the server which program is accessing the server. This field is frequently used by servers which auto-generate pages to decide what features to include in a page. It is also used by statistics programs to decide which browsers are accessing a page. The third parameter tells the server which page contained the link to the page.

Three standard HTTP methods have been defined:

GET	- Fetches the desired address
HEAD	- Gets information about the address
POST	- Posts information to the server address.

Requests may include a number of parameters (headers), such as:

Host: opera.nta.no
- Tells the server by which name it is being accessed.

Cookies: name=opera
- Values previously stored by the server on the client machine.

If-Modified-Since: date
- Requests that the file should only be returned if it has been changed.

Referrer: http://opera.nta.no
- Tells the server from which page it is being accessed.

User-Agent: Opera 2.12
- Tells the server which program is accessing it.

With the advent of HTTP 1.1 there have been some changes to the parameters and the way the server works. The client can now request that the server keeps the connection alive so the long connection times can be avoided when fetching multiple documents from the server. The client must then tell the server when no more requests will be coming.

One HTTP request gets only one document, while a typical Web document is made up of an HTML document and a number of images, sounds, animation, etc. This means that after fetching the first file (most often an HTML document) the Web client must request more files to be able to show the finished document. Modern browsers will fetch multiple files at the same time sending multiple HTTP requests to the server or servers, as a Web document may be created from files on multiple servers.

HTML, the language of choice

> *There is a common perception that HTML is a programming language and is therefore reserved for the technically literate. This perception is endorsed (and often promulgated) by the growing army of HTML programmers, who see the new World Wide Web marketplace as an employment opportunity. And it's an excellent opportunity for these people. But make no mistake, calling HTML a programming language is like calling a janitor a sanitation engineer – it sounds nice but anyone can learn to scrub a floor with a little knowledge and effort.*
>
> Tom Savola

HTML is the document format used on the Web and is one of the reasons why the Web has become so popular. The original versions of HTML were very simple, allowing users to quickly make their own pages with information. The documents are hypermedia documents, meaning that documents are connected through links in the documents. By clicking a link, the user can move from one document to another. The documents can be viewed on just about any machine platform. Images and such cannot be displayed on all machines, but the generic contents can be viewed and links can be followed. This also means that HTML is a means to allow handicapped users, the blind, deaf and others to have equal access to information as the information can be formatted to their requirements. The whole point is that HTML is a logical language. It describes the contents and the logical structure of the documents, but leaves the formatting to the presentation tool, the browser. The document may include formatting hints, though.

The complexity of HTML has been steadily increasing with the commercialisation of the Web. At the same time, HTML has been moving away from its roots becoming more and more like a normal document format. The announced Netscape variant even includes exact positioning and multiple layering making the documents even less portable as it is much more difficult to decide how to present this kind of document on non-graphical devices.

HTML is no longer a single language or a standard. There are a number of variations which makes it even more difficult for the designer. Many designers have fallen for some of the new proprietary extensions, making their documents unusable on browsers that do not support those extensions. In this way they have made their documents unusable by the masses as most users do not use the latest versions of the programs and others use programs that are not compatible with the proprietary extensions.

Below is a description of the various HTML versions and what they have contributed to the HTML language.

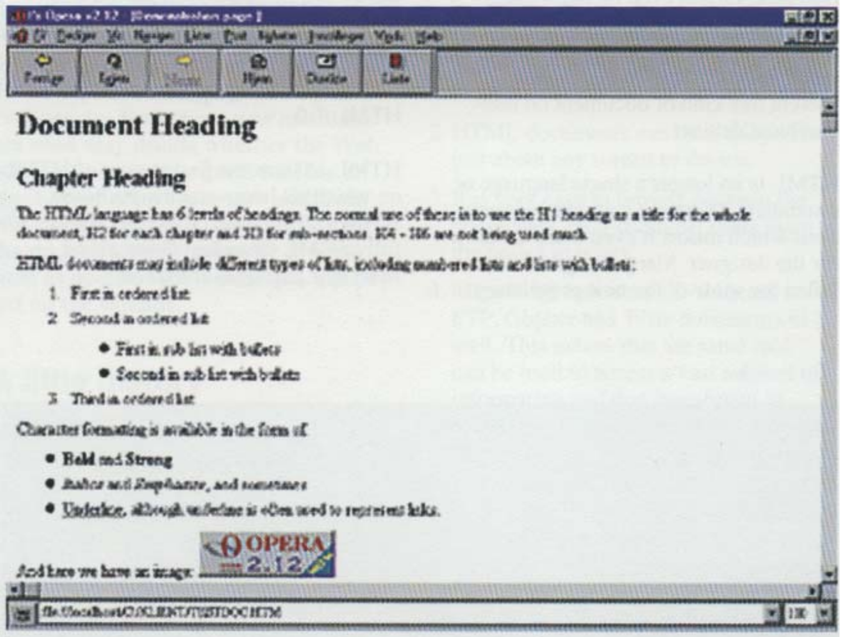
HTML 1.0

HTML 1.0 was the first variant of HTML on which all later versions are based. HTML 1.0 is based on SGML and is fully SGML (Standard Generalised Markup Language) compliant.

```
<HTML>
<HEAD>
<TITLE>Demonstration page</TITLE>
<BODY>
<H1>Document Heading</H1>
<H2>Chapter Heading</H2>
The HTML language has 6 levels of headings. The normal use of
these is to use the H1 heading as a title for the whole docu-
ment, H2 for each chapter and H3 for sub-sections. H4 - H6 are
not being used much.
<p>
HTML documents may include different types of lists, including
numbered lists and lists with bullets:
<OL>
<LI>First in ordered list
<LI>Second in ordered list
<UL>
<LI>First in sub list with bullets
<LI>Second in sub list with bullets
</UL>
<LI>Third in ordered list
</OL>
Character formatting is available in the form of:
<UL>
<LI><B>Bold</B> and <STRONG>Strong</STRONG>
<LI><I>Italics</I> and <EM>Emphasize</EM>, and sometimes
<LI><U>Underline</U>, although underline is often used to rep-
resent links.
</UL>
<P>And here we have an image:
<IMG SRC=os212.gif ALT="Opera download button 2.12">
</BODY>
</HTML>
```

Figure 1 Example HTML 1.0 document

HTML 1.0 documents can be easily displayed on any kind of device. The document format includes normal, text, headers, images, lists of different types and a few formatting hints such as *bold* and *italic* although purists will expect one to use more logical definitions such as *strong* and *em* as these indicate the meaning of the text, not the presentation. Figure 1 shows an example HTML document and Figure 2 shows the result as it is displayed on a graphical browser. Figure 3 shows the same result without graphics.



Document Heading

Chapter Heading

The HTML language has 6 levels of headings. The normal use of these is to use the H1 heading as a title for the whole document, H2 for each chapter and H3 for sub-sections. H4 - H6 are not being used much.

HTML documents may include different types of lists, including numbered lists and lists with bullets:

1. First in ordered list
2. Second in ordered list
    - First in sub list with bullets
    - Second in sub list with bullets
3. Third in ordered list

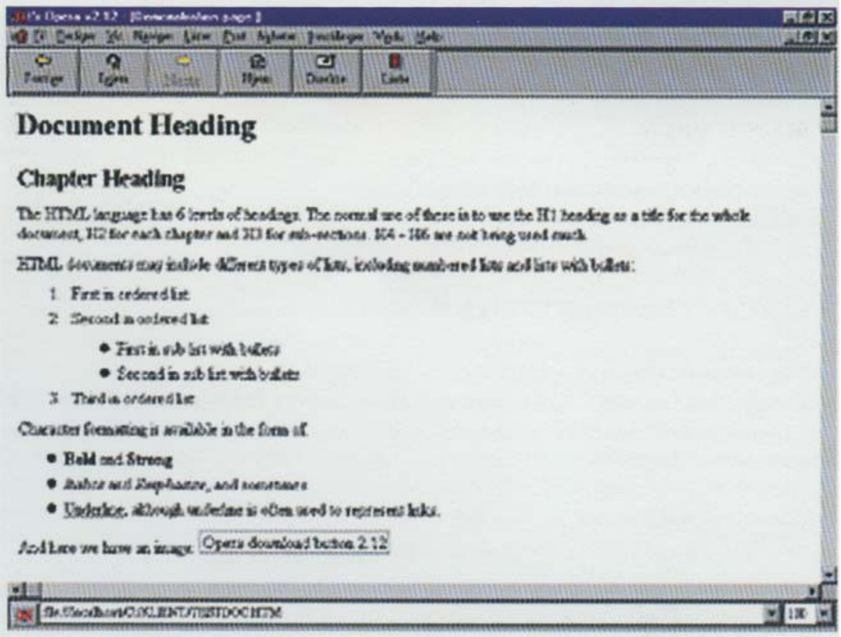
Character formatting is available in the form of:

- **Bold and Strong**
- *Italics and Emphasize, and sometimes*
- Underline, although underline is often used to represent links.

And here we have an image:

file://localhost/C:/CLIENT/TESTDOC.HTM

Figure 2 Example HTML 1.0 document with images turned on



Document Heading

Chapter Heading

The HTML language has 6 levels of headings. The normal use of these is to use the H1 heading as a title for the whole document, H2 for each chapter and H3 for sub-sections. H4 - H6 are not being used much.

HTML documents may include different types of lists, including numbered lists and lists with bullets:

1. First in ordered list
2. Second in ordered list
    - First in sub list with bullets
    - Second in sub list with bullets
3. Third in ordered list

Character formatting is available in the form of:

- **Bold and Strong**
- *Italics and Emphasize, and sometimes*
- Underline, although underline is often used to represent links.

And here we have an image: Opera download button 2.12

file://localhost/C:/CLIENT/TESTDOC.HTM

Figure 3 Example HTML 1.0 document with images turned off

HTML 2.0

As HTML started to be used, a lot of new ideas were created as to how to make the documents more interactive. The most interesting addition to the HTML language in version 2.0 is the forms elements. Forms enable the designer to have users fill in information so that a program on the server can perform searches in databases based on the information provided by the user (the most common use of forms) or ask the user for information in general. Figure 4 shows the HTML code for a simple form. Figure 5 shows the result as shown by a graphical browser.

The second major addition is the ISMAP element which allows the designer to have elements with multiple active areas. This addition makes HTML less portable as the co-ordinates are sent to the server which in turn returns the correct page. The browser does not know what the map really contains and must present it graphically for the user to know what it contains.

Spyglass Mosaic HTML extensions

Spyglass, the company that NCSA chose to represent the commercial rights to their Mosaic browser has made one valuable addition to HTML, rectifying the problems created by the addition of ISMAP. Client Side ISMAP enables the use of images with multiple hot spots, but the description of the image hot spots is stored in a document actually fetched by the browser. This makes it possible for the browser to display this information in a way readable by the user, even without the use of graphics. For this to be possible, though, the designer must have put information in the document to describe the various parts of the image. Not all designers do this.

HTML 3.0

HTML 3.0 was the working title of a new version of HTML. This version had a number of interesting and exciting extensions, but it never materialised. Instead HTML 3.2 was released. A number of the ideas in HTML 3.0 have been adopted by the Netscape HTML and included in the final HTML 3.2, which is basically HTML 2.0 + a few Netscape extensions.

Tables were in the HTML 3.0 working draft, but the table draft kept changing. Netscape implemented their own table standard which is based on the HTML 3.0 table standard, but is not totally compatible. Tables are a powerful tool for document design as it makes it possible for the designer to place text in columns etc. Sadly, the table design has been much misused and the result is that the final HTML is often less portable. Examples include pages that only use part of the screen on some machines, while users need to scroll horizontally on other machines.

Alignment of images was another addition in HTML 3.0. Images could now be placed either to the left or to the right of text, but not in the middle, although that kind of result could be achieved through the use of tables.

Background images enhance the document design even more. The designer can choose an image or a colour to make the document more exciting. Given that the designer avoids mistakes such as placing green text on a red background (and thus giving colour-blind readers a problem) this is a really useful feature.

Netscape HTML

Netscape, the current leader in the browser market, did not want to wait for the standardisation organisations to come up with changes in HTML. They started making their own extensions. Most of these extensions give the designer more power at the expense of portability.

Frames are a way to divide documents into sections. The sections are independent and can be scrolled independently. Frames are loved by a number of designers and some designers have gone so far as to only make pages with frames, promptly ignoring more than 50 % of the market. In addition, they urge users to buy a frames capable browser. Frames make documents even less portable.

Another new extension is font size and colour. This extension gives the designer even more flexibility to make exciting documents. If the designer uses these extensions and ignores the normal header tags, it is more difficult for programs to extract header information for use by programs presenting information for the blind. The logical structure is lost.

Microsoft HTML

Microsoft's strategy when it comes to the Internet is in their own words to "embrace and enhance". This is what they have done. They are a great supporter of the standardisation process, and embrace standard additions, but at the same time they add their own extensions. Most of these extensions do in no way destroy the structure of the documents, but just add more multi-media features. Microsoft's non-HTML extensions, such as ActiveX are a bigger problem.

Microsoft's extensions include inline video support for their own video format, background sound, table background colours, scrolling text, fixed background that does not scroll and floating frames, that is frames that can be moved in location inside a document. Microsoft also added the font face extension. Many of these extensions are now supported by other browsers.

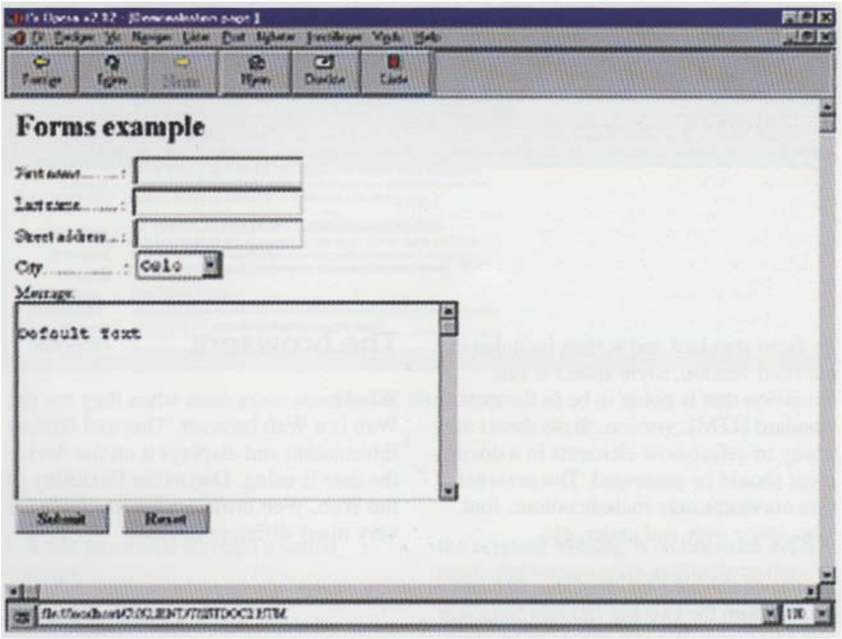
HTML, the future

The standardisation process continues, but again it is mostly based on implementations. The standardisation community suggests something and if Netscape and Microsoft implement it, it becomes a

```
<HTML>
<HEAD>
<TITLE>Demonstration page</TITLE>
<BODY>
<H1>Forms example</H1>
<FORM method=GET action="http://www.opera-
software.com/cgi-bin/none.cgi">

First name ......: <INPUT TYPE=text
                    NAME=firstname><BR>
Last name .......: <INPUT TYPE=text
                    NAME=lastname><BR>
Street address ..: <INPUT TYPE=text
                    NAME=street><BR>
City ...........: <SELECT name = city>
<OPTION SELECTED VALUE="OSLO">Oslo
<OPTION VALUE="BERGEN">Bergen
<OPTION VALUE="TROMSØ">Tromsø
</SELECT><BR>
Message:<BR>
<TEXTAREA NAME = textmessage ROWS=10
COLS=40>
Default Text
</TEXTAREA><BR>
<INPUT TYPE=submit>
<INPUT TYPE=reset>
</FORM>
</BODY>
</HTML>
```

Figure 4 Example forms document



Forms example

First name ......:

Last name ......:

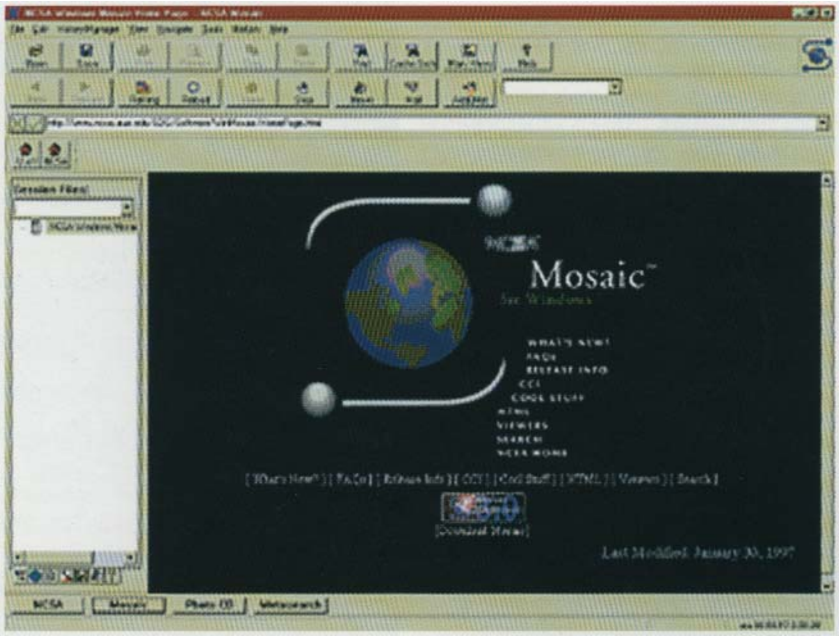
Street address ..:

City ....: Oslo

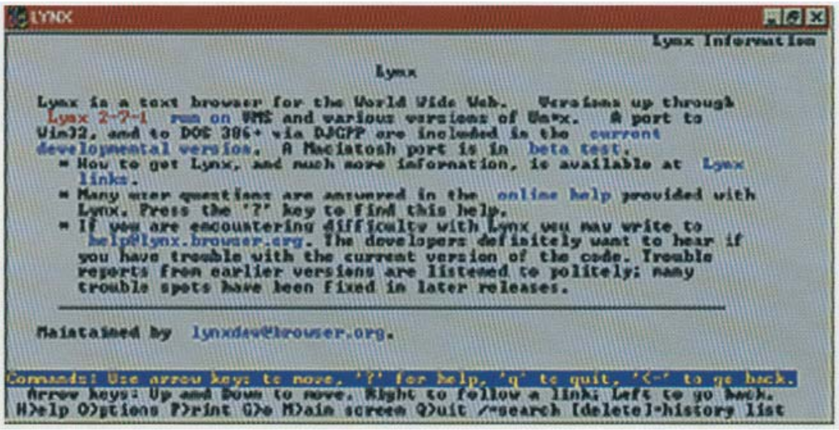
Message:

Default text

Figure 5 Example forms document in graphical browser



NCSA Mosaic



```
                                                        Lynx Information

                                   Lynx

   Lynx is a text browser for the World Wide Web.   Versions up through
    Lynx 2-7-1  run on VMS and various versions of Un*x.   A port to
   Win32, and to DOS 386+ via DJGPP are included in the  current
   developmental version.  A Macintosh port is in  beta test.
      * How to get Lynx, and much more information, is available at  Lynx
        links.
      * Many user questions are answered in the  online help provided with
        Lynx. Press the '?' key to find this help.
      * If you are encountering difficulty with Lynx you may write to
         help@lynx.browser.org. The developers definitely want to hear if
        you have trouble with the current version of the code. Trouble
        reports from earlier versions are listened to politely; many
        trouble spots have been fixed in later releases.
     ____________________________________________________________________

   Maintained by  lynxdev@browser.org.

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
  Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

Lynx

de-facto standard and is then included in the final version. Style sheets is one extension that is going to be in the next standard HTML version. Style sheets are a way to define how elements in a document should be presented. The presentation attributes may include colour, font size, space over and under, etc.

However, the greatest changes can be expected from the two big fighters Netscape and Microsoft. We can expect even more changes that make HTML less portable.

The browsers

What most users meet when they use the Web is a Web browser. This tool fetches information and displays it on the device the user is using. Due to the flexibility of the Web, Web browsers are available for very many different devices.

NCSA Mosaic

Mosaic was one of the first Web browsers and the browser that made the Web very popular in the first place. This Web browser really showed what was possible to achieve. This browser was originally available on UNIX machines, but was quickly ported to a number of other platforms, although the quality of Mosaic on other platforms never reached that of the UNIX version.

The Mosaic code has been licensed to a number of companies and many browsers are based on the original code. The original Mosaic is still around, and has lately shown some innovative features.

Lynx

Lynx is a simple non-graphical browser. It is widely used on UNIX machines all around the world. Lynx is a favourite among users that are more concerned about the contents of documents than the layout. A number of solutions have been made with Lynx to enable handicapped users to reach the Web.

Cello

In the early days of the Internet, Cello was one of the contenders for the throne on Microsoft Windows based computers. Cello was a simple Web browser, but compared to the Mosaic available on Windows at the time, it was quite good. Sadly, Cello development was not continued, although there were rumours that rights to the code had been sold to a commercial vendor.

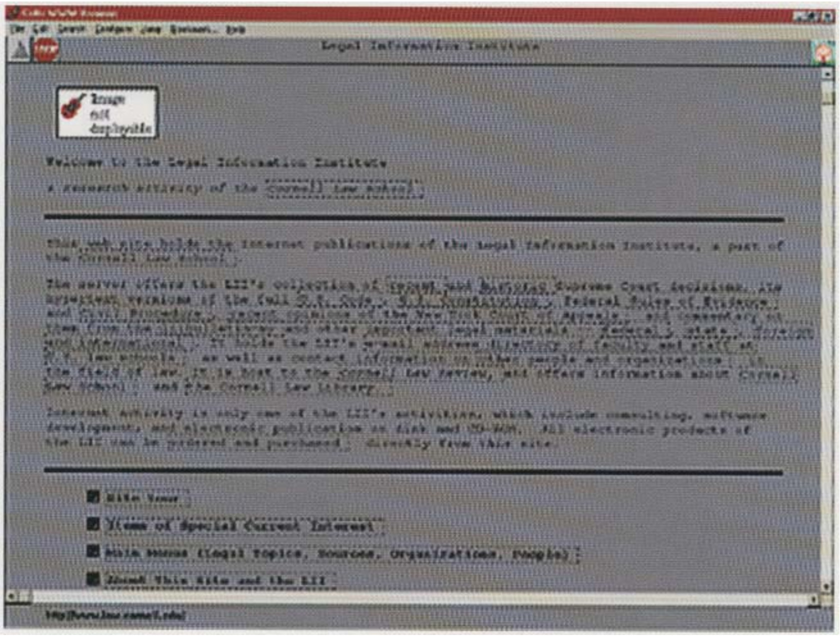
Mosaic variations

Spyglass got the job from NCSA to commercialise Mosaic and they have done a good job of making the program available to other parties. Mosaic has therefore been the base of a number of browsers, including offerings from Spry, Quarterdeck and Microsoft. Of these, most are not being actively developed anymore. The exception is Microsoft Explorer. Microsoft spent a long time understanding that Internet was something to take into consideration. When they finally understood, they licensed the rights to the Mosaic code and started working on improvements. Microsoft also use their unique position in the market to try to kill off Netscape, their main competitor. Microsoft has gone as far as to give their product away for free and even including free services as well. This practice has been severely criticised in a number of articles written by people who do not like this form of dumping.

Netscape

Netscape Communications is a company that was founded by the main developer of Mosaic, Mark Andreassen and Jim Clark, former head of Silicon Graphics. Most of the developers of NCSA Mosaic joined Netscape to develop Netscape Navigator, the most popular Web browser at the time of writing.

Netscape has been technology driven and due to their position they have been able to dictate the directions of the development. This has led to a number of extensions to the HTML language that are far away from the original design and that make HTML much less scaleable. Netscape is now in a feature war with Microsoft. This has led to both browsers becoming fat and slow.



Legal Information Institute

Welcome to the Legal Information Institute

A research activity of the Cornell Law School

This web site holds the Internet publications of the Legal Information Institute, a part of the Cornell Law School.

The server offers the LII's collection of recent and historic Supreme Court decisions, its hypertext versions of the full U.S. Code, U.S. Constitution, Federal Rules of Evidence and Civil Procedure, recent opinions of the New York Court of Appeals, and commentary on them from the [illegible], and other important legal materials — federal, state, foreign and international. It holds the LII's e-mail address directory of faculty and staff at U.S. law schools, as well as contact information on other people and organizations in the field of law. It is host to the Cornell Law Review, and offers information about Cornell Law School and the Cornell Law Library.

Internet activity is only one of the LII's activities, which include consulting, software development, and electronic publication on disk and CD-ROM. All electronic products of the LII can be ordered and purchased directly from this site.

- Site Tour
- Items of Special Current Interest
- Main menus (Legal Topics, Sources, Organizations, People)
- About This Site and the LII

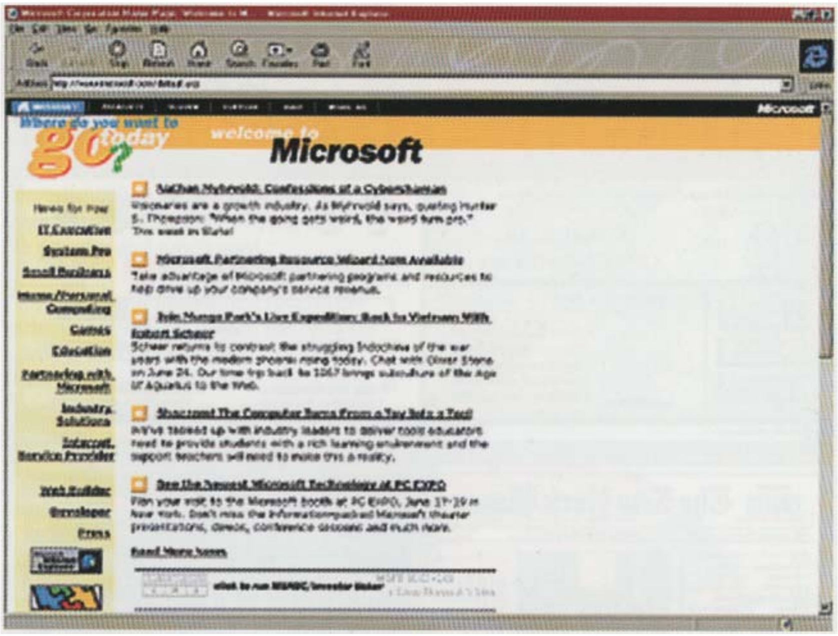
http://www.law.cornell.edu/

Cello

Opera

With Netscape and Microsoft in a feature war to the bitter end, a market opened for someone doing something different. Someone that puts the user in focus and focuses on solving the main problem with using the Internet today: speed.

Opera Software is the company that has decided to take up the fight with Microsoft and Netscape. Opera, the company's offering, takes up a fraction of the size of Netscape and Explorer, runs on minor PCs and is faster. Opera also allows the user to tailor its looks and the presentation of information much more, thus making the Internet available for more users.


Microsoft Corporation Home Page: Welcome to M... - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
Print
Font
Address
Links
Microsoft
Where do you want to go today?
welcome to
Microsoft

News for You
IT Executive
System Pro
Small Business
Home/Personal Computing
Games
Education
Partnering with Microsoft
Industry Solutions
Internet Service Provider
Web Builder
Developer
Press

Nathan Myhrvold: Confessions of a Cybershaman
Visionaries are a growth industry. As Myhrvold says, quoting Hunter S. Thompson: "When the going gets weird, the weird turn pro." This week in Slate!

Microsoft Partnering Resource Wizard Now Available
Take advantage of Microsoft partnering programs and resources to help drive up your company's service revenue.

Join Mungo Park's Live Expedition: Back to Vietnam With Robert Scheer
Scheer returns to contrast the struggling Indochina of the war years with the modern phoenix rising today. Chat with Oliver Stone on June 24. Our time trip back to 1967 brings subculture of the Age of Aquarius to the Web.

Snapshot: The Computer Turns From a Toy Into a Tool
We've teamed up with industry leaders to deliver tools educators need to provide students with a rich learning environment and the support teachers will need to make this a reality.

See the Newest Microsoft Technology at PC EXPO
Plan your visit to the Microsoft booth at PC EXPO, June 17-19 in New York. Don't miss the information-packed Microsoft theater presentations, demos, conference sessions and much more.

Read More News

click to run MSNBC/investor ticker

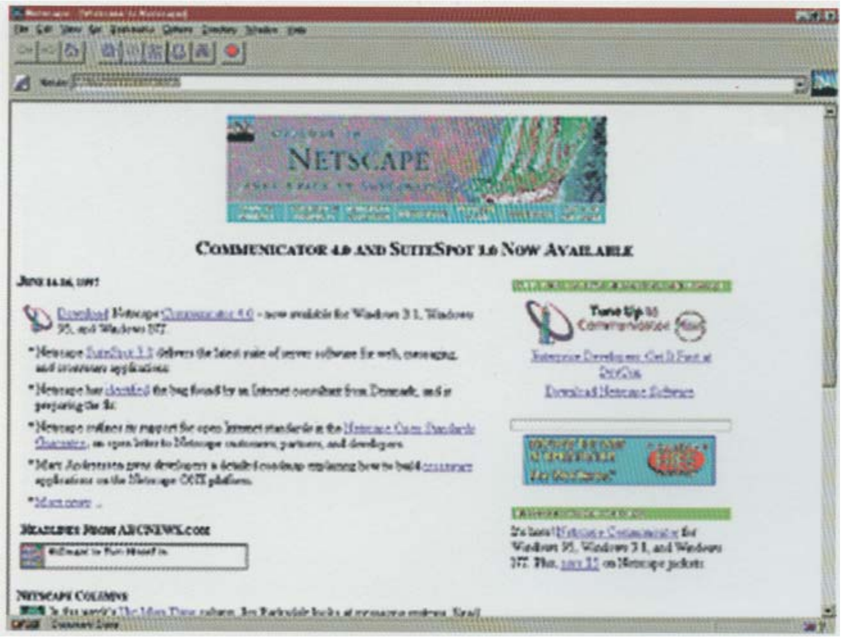
Mosaic variations

The servers

Although the client software is what the user meets when accessing the Internet, it takes two to dance. The server is the dance partner. The server is a much simpler piece of program. It receives requests from the client and responds with one of the following:

1. Error message indicating that the file was not found or is not available
2. The file
3. A file generated through a called script.

The first major server was made by NCSA, the same organisation that made the original Mosaic Web browser. NCSA made the source code available to the general public and most, if not all, Web servers on the market today are based on the NCSA code.



Netscape

Opera

The second major player at the start of the Web adventure was CERN. Not to be outdone, they made the CERN Web server, which is still a major player. CERN introduced a number of new features, the most useful one being proxy and caching. Through the use of a proxy server companies can more easily control traffic as all requests from employees is done through the server. This keeps the complexity of firewall solutions to the minimum. The firewall need only know a handful of machines. The cache bit is another bonus as pages that are frequently fetched are being received much faster. The downside is that new pages often tend to take longer to download.

Soon, commercial companies started to make Web servers with Netscape at the forefront. Netscape now has a number of different servers gathering for different needs. Microsoft, not to be outdone, has released its own servers that compete with Netscape. Other players include O'Reilly with their Website Server and a number of smaller players.

The most popular Web server, however, is Apache. Apache, like the NCSA and CERN servers, is free. It is also based on the original NCSA server.

The Netcraft Web Server Survey (http://www.netcraft.com/survey/) is a survey of Web Server software usage on Internet connected computers. They collect and collate as many host names providing an HTTP service as they can find, and systematically poll each one with an HTTP request for the server name. In the May 1997 survey responses from 1,044,163 sites were received. Figure 6 shows relative percentage of the market for the biggest Web servers.

Another source for statistics is WebCrawler (http://www.Webcrawler.com/WebCrawler/Facts/Servers.html). Their results differ somewhat from the Netcraft results (Figure 7). Interestingly, they also keep a log of servers based on platform. The log shows that UNIX servers still dominate the market.

The servers have started to include a number of extra functions. Early servers had a scripting facility which made it possible to make information on various databases available through the server. Newer servers include database handling, simplified set-up and security.

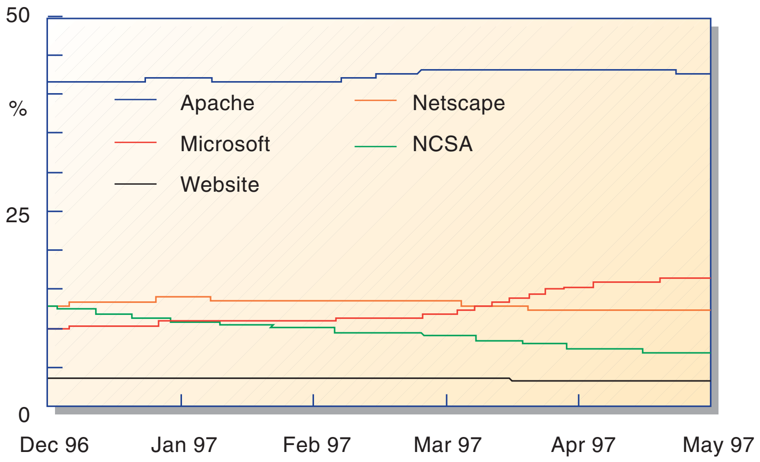


Figure 6 Netcraft Web Server Survey results

Security

Information and information requests on the Internet are mostly sent unencrypted. This means that anyone with a little knowledge of how the Internet works can monitor communication. This was fine as long the Internet was in use for academic purposes, but as the Internet is now being used for internal company information and for buying and selling, security has become a big issue.

E-mail is still mostly insecure. Although there are solutions, none are in wide use, meaning that they do not become used exclusively. Also, some of them are difficult to use and require special programs beside the e-mail program.

The Web is a little more secure. Netscape made their own security standard, SSL (Secure Socket Layer). The security with this standard is reasonable for the use that it is intended. However, due to US restrictions, Netscape, Microsoft and other US companies cannot export secure versions of SSL, only part secure versions. This means that non-US countries using software from the two major companies are using non-secure software.

In addition to SSL, there is one other standard, S-HTTP (Secure HTTP), endorsed by the World Wide Web consortium. This is not in wide use.

In time we will see more and more card based solutions for Internet Security. Opera Software already has a solution based on smart-cards which gives very high security on information inside a company and for the selling of information on the Internet.

Netscape Plug-ins

> *Netscape Navigator plug-ins are dynamically loaded code modules that become part of the browser's code path. That is, after the plug-in is loaded, it becomes a direct part of the browser's code. This technique provides the best possible speed, but it lacks in security and platform independence.*
>
> Zan Oliphant

Netscape plug-ins are dynamically loaded code modules that become part of the browser's code. Although this is originally a Netscape only thing, other browser vendors have been adding support for Netscape plug-ins as well.

Plug-ins are a nice way to add functionality to a browser as they can be well integrated into the browser, and as they are most often written in C/C++ they can also be quite fast. Plug-ins are especially useful in an Intranet setting as it is easy to make sure that all users have the relevant plug-in. On the Internet, plug-ins are less useful as most users do not want to download a new code module every time they go to a new site. Instead a small number of plug-ins are popular and much used.

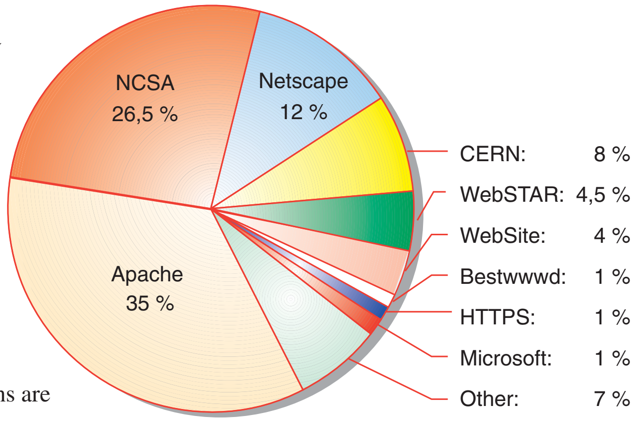


Figure 7 WebCrawler Server Statistics

Plug-ins are all placed in a special directory for plug-ins. When the browser starts, it looks in this directory to see what plug-ins are available. Each plug-in is defined to handle one or more MIME types and will only be loaded when this MIME type is found in a document. This means that the plug-in is not wasting memory as long as no document has been loaded that uses it.

Many different types of plug-ins are available. Here is a short list to indicate what they can do:

- Fractal Image Viewer by Iterated Systems
- Acrobat document format viewer by Adobe
- Shockwave multimedia viewer from Macromedia

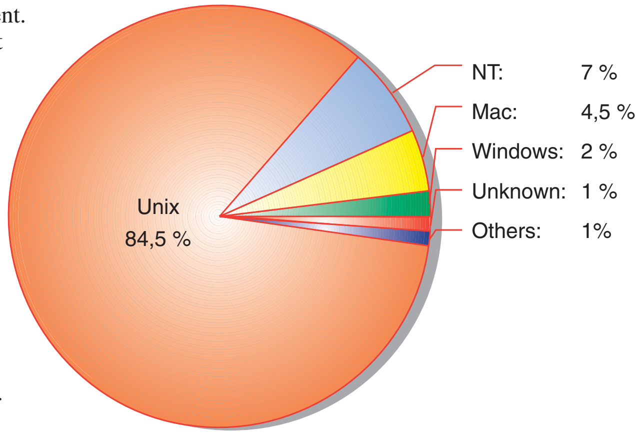


Figure 8 WebCrawler Server Statistics by Platform

- Live3D VRML (Virtual Reality) viewer from Netscape
- RealAudio real time audio player from Progressive Networks
- ActiveX from Ncompass (more on the ActiveX technology later).

Sadly, Netscape plug-ins are both program and platform dependent. This means that the scope and flexibility of the Web can be lessened. Inside an Intranet, this is not really a problem, but this does mean that documents using plug-ins will have a smaller potential user base than other documents. Some information providers are not aware of this and decide that their users will be running their recommended software and do not have any kind of physical handicap. This also means that any serious company or institution with the general public as a target should not be using plug-ins on their Web sites.

JAVA

> *Forrest Gump might say that JAVA is as JAVA does. JAVA is a programming language, a runtime system, a set of development tools, and an application programming interface (API).*
>
> Jamie Jaworski

To most users JAVA is that hype thing that makes Web pages look fancy and slows down the Internet to a crawl. To a certain extent they are right, although users often mistakenly take JavaScript scripts or even animated figures to be JAVA. To others JAVA is the solution to a world of problems, through its unique, powerful solutions. To security experts JAVA is a nightmare.

```
<HTML>
<HEAD>
<TITLE>JavaScript generated HTML</TITLE>
</HEAD>
<BODY>
<H1>JavaScript generated HTML</H1>
<SCRIPT>
var now = new Date();
document.write("Today it is " + now);
</SCRIPT>
</BODY>
```

Figure 9 JavaScript

What makes JAVA so special is that it is purposed to be a solution for a lot of things. Some of this is unrealistic, but JAVA may have a great future. This is why:

- JAVA is platform-independent. This means that JAVA code in theory can be run on any platform, given that a JAVA virtual machine is implemented on the platform.
- JAVA runs programs off the net. This means, in theory for now due to the lack of speed on the net, that programs can be loaded off the net and run on machines with hardly any disk space. Distribution and maintenance of software becomes much easier.
- JAVA tries to be a secure solution. JAVA programs are not allowed to mess things up on the user's machine. In fact, JAVA has succeeded quite well in this department, although security experts have pointed out various flaws and holes.

JAVA programs are written in the JAVA programming language. The programming language is quite extensive. Programming JAVA can be compared to programming in C++. This means that programming JAVA is not for basic users, although tools are showing up on the market that ease the creation of JAVA programs and applets (a JAVA applet is a JAVA program running inside a browser). When the programmer is content with his program, the program is converted to bytecode. This form is more compact than the original file and is platform independent. The bytecode can be run on a JAVA virtual machine (JVM), implemented on a series of different platforms. In practice, a lot of the JVM implementations compile the JAVA program before running it, instead of interpreting it line by line.

JAVA is in many ways a great thing. It opens up a lot of possibilities and is, in theory, an open standard. However, with JAVA a number of ready-made modules are delivered that can be used by the programmer. Netscape and Microsoft, as the biggest actors in the market, deliver different libraries making JAVA code less portable.

JavaScript

> *Ask any JavaScript programmer: the most notable feature of JavaScript (Navigator 2.0 and early betas of 3.0 at least) is its bugs. And the most common experience of JavaScript programmers is frustration.*
>
> David Flanagan

JavaScript Is Netscape's scripting language. This is a very powerful language that allows the document to interact with the browser and the user in a number of ways. Sadly, the most common use of the language today is to scroll text in the status area. This is a simple thing to do, while more elaborate programming is required to do anything really useful. The problems with bugs and the fact that many users will continue to use early versions of Netscape is another problem.

JavaScript is found in two versions, one running on the client and one running on the server. We will concentrate on the client side JavaScript, as this is the one that is most used.

JavaScript is a full programming language with variables, objects, loops and functions. It has a number of built in types such as number, Boolean and string, but it is also untyped. This means that the variables need not be declared before they are used and that values can usually be automatically converted to any other type.

JavaScript is embedded in the HTML document. Figure 9 shows a simple script inside a very simple document.

JavaScript has a number of ready made objects that the programmer can use to interact with the program. The following objects are available:

- The self, window, parent and top objects give access to the various windows and frames.
- The Navigator object gives access to information about the browser, the plug-ins and MIME types it supports.
- The frames object array.
- The location object allows the programmer to check what location is being displayed and to display another document.
- The history object gives the programmer access to the history, allowing jumps.
- The document object.
- The packages object.

JScript is Microsoft's implementation of JavaScript. It implements a subset of the full JavaScript. This means that the programmer has to know which parts of JavaScript JScript implements as to ensure that scripts work with both of these programs. In addition, the programmer has to remember to check his script on different versions of these programs as the scripting language has been changing from version to version. Some of the changes mean that older scripts may not work on newer versions of the browsers.

VBScript

As an alternative to JavaScript, Microsoft made VBScript (Visual Basic Script). VBScript is based on the Microsoft Visual Basic language and should be easy to use for programmers familiar with this programming language. As VBScript is only supported by Microsoft it is unlikely that it will become widely used unless Microsoft succeed in killing off its competitors.

ActiveX

> *ActiveX is anything that Microsoft can give to the developer community that will bring about the integration of the computer desktop environment with the environment that makes up the Internet and its myriad resources and potential, all the while leveraging existing developer investment in Windows technology.*
>
> Eric Tall and Mark Ginsburg

ActiveX is another Microsoft invention. Previously referred to as OLE controls, ActiveX modules are program modules that slot into programs to add functionality. ActiveX is currently only supported by Microsoft, but others may follow. ActiveX can in many ways be compared to plug-ins and is also seen as an alternative to JAVA, although much less secure as ActiveX modules have full access to the operating system and files.

Operating systems – final words

> *Microsoft has long argued that the setting of pre-emptive standards by a regulatory body is often premature in the technological industry. One of the company's white papers speculates that if the government had intervened in the PC arena, we would now be standardised on TRS-80s.*
>
> Eric Tall and Mark Ginsburg

The Internet war is not a new war. It is in many ways a continuing war for being the leading distributor of computer software solutions. Microsoft has a leading position through their operating systems, MS-DOS and Windows. The threat to Microsoft is that Internet software can in fact run on operating systems and through use of other software that is not delivered by Microsoft and this may be enough for users not needing Microsoft's other applications. This is why Microsoft is so desperate to get users and developers to use their tools and systems that they are willing to give away software for "free". In addition, they offer software developers "standards" which enable them to deliver solutions faster on Microsoft systems. This is in accordance with Microsoft's earlier gameplay, where the holding of standards is their main playing card.

The downside for the user is that given that Microsoft wins this war we may end up with one software company delivering the operating system, network, security and many other components of not only computers, but loads of other devices.

Microsoft has some opposition, mainly IBM, SUN, Oracle and Netscape. This alliance is working hard on the JAVA standard and hope that through this they can push Microsoft off its throne. In addition, smaller companies, such as Opera Software, run around in this giant playground and may just grab a sizeable part of the pie as the giants concentrate on slaying each other.

Terms

Firewall	Solution to keep unwanted users out and possibly filter outgoing traffic as well.
FTP	File Transfer Protocol
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
MIME	Multipurpose Internet Mail Extensions
SGML	Standard Generalised Markup Language
WWW	World Wide Web

Bibliography

Chappell, D. *ActiveX and OLE : A Guide for Developers & Managers.* Washington, Microsoft Press, 1996.

Flanagan, D. *JavaScript : the definitive guide.* Cambridge, O'Reilly, 1996.

Jaworski, J. *JAVA Developer's Guide.* Indianapolis, Sams.net, 1996.

Oliphant, Z. *Programming Netscape Plug-Ins.* Indianapolis, Sams.net, 1996.

Roberts, D. *Internet Protocols : Handbook.* Scottsdale, Coriolis Group Books, 1996.

Savola, T. *Using HTML : Special Edition.* Indianapolis, Que Corporation, 1995.

Tall, E, Ginsburg, M. *Late Night ActiveX.* California, Ziff-Davis Press, 1996.

Jon S. von Tetzchner is CEO of Opera Software. He worked two years at Telenor R&D on Internet and Intranet issues, and then founded Opera Software in 1995 with Geir Ivarsøy.

email address:
jons@nta.no

Opera Software can be found at
http://www.operasoftware.com

IPv6 overview

ESPEN KLOVNING

This article presents a short overview of the Internet Protocol version 6 (IPv6) which eventually will replace the existing version. Some of the most important features of IPv6 including Quality of Service support, mobility, security and multicasting are described along with the built-in transitional support to make incremental transition to the new Internet Protocol possible.

1 Introduction

The number of Internet users and the Internet traffic is increasing at an exponential rate. Today, one can foresee future IP based services which would have been impossible to imagine just a few years ago. The next generation of services will include network based entertainment and electronic commerce. Other services will be nomadic computing and possibly device control. It is even possible that household articles will be networked during the next decade. The requirements to the Internet protocol suite will be tough.

Version 4 of the Internet Protocol [1, 14] has been a tremendous success, integrating subnetworks to a world spanning Internet. However, during the last decade it has become apparent that IPv4 eventually will run out of address space due to its 32 bit addresses and rigid addressing format. The introduction of Classless Inter-Domain Routing (CIDR) [12] discards the different IPv4 address classes and routes based on a longest prefix matching scheme. CIDR has extended its lifetime, but with the current Internet growth a new Internet Protocol will be needed soon.

In addition to the address space problem, the functionality of IPv4 would not be able to meet the requirements of the next generation of applications and Internet services. Some of the areas where IPv4 will fall short of the requirements are within security, nomadic computing, mobility, multicast and Quality of Service issues.

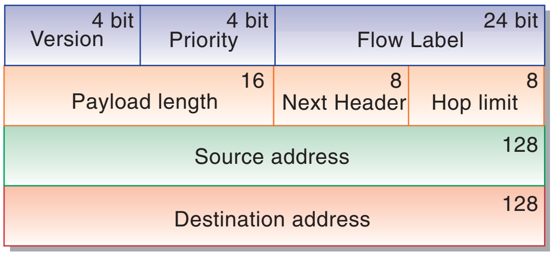


Figure 1 IPv6 packet format

In 1992, the Internet community came up with several different proposals for the next generation IP, and the final specification denoted Internet Protocol version 6 was finished in 1995 [2, 9, 10].

This article is outlined as follows: In Section 2 the protocol formats of the IPv6 protocol are described, while its functionality is described in Section 3. The important transition mechanisms which include dual protocol stacks and tunneling of IPv6 over IPv4 are described in Section 4. The article closes with some concluding remarks.

2 Internet protocol version 6 – formats

2.1 IPv6 packet header format

Based on the experience with the IPv4 packet header and the new services which are needed, IPv6 includes some new protocol fields as illustrated in Figure 1. The most significant change is that the source and destination addresses are extended from 32 to 128 bits [3]. The Time-to-live field in IPv4 has been replaced with a Hop-limit field in IPv6 which is how most IPv4 routers have interpreted it anyway. The fragmentation and option fields have been moved out of the IPv6 header. They are now part of the extension headers which is the topic of the next subsection. The type-of-service field has been replaced by two fields, a flow label and a priority field. The protocol field of IPv4, which has been used to indicate the upper layer protocol, has been modified. The next header field in the IPv6 points to the next extension header, if present. If no extension headers are present, this field will include a protocol type identification similar to IPv4. The alignment of the protocol header has been changed from 32 to 64 bit, which is important for efficient routing operation and other IPv6 protocol optimization efforts.

The increase in the source and destination addresses to 128 bit combined with the restructuring of the IP header lead to a doubling of the IP header from 20 to 40 bytes. However, the fact that all the headers are located in a fixed-size IPv6 header will decrease the routing processing cost.

In [13] it is shown that the IPv6 addressing hierarchy is efficient compared to the hierarchies used for the phone networks. The evaluation shows that a conservative estimate suggests that the IPv6 addressing solution permits 1564 IPv6 addresses per square meter of the earth's surface. Initially, 15 % of the IPv6 address space is assigned, while 85 % is reserved for future use.

One problem with the modification of the IPv6 header is that higher layer protocols which use for example the IPv4 source and destination addresses to compute their checksums must be modified accordingly. In addition, computations in higher layers which use the IP header size have to be modified as well. This information is for instanced used in BSD based TCP/IP implementations for efficiency reasons [21].

2.2 IPv6 address formats

Unicast address

A unicast address identifies a single interface of an IPv6 node. This address type can be structured in several ways, e.g. provider-based global, link-local, site-local, IPv4 compatible IPv6 address and loopback address.

The *provider-based global* unicast address offers global addressing throughout the entire Internet. The *link-local* address can be used for addressing within a single link or subnet and cannot be modified to be used in global addressing. The *site-local address* can be used for addressing within a single site. This address type can easily be integrated with the global addressing scheme by simple reformatting. The upper part of this address can be replaced by a global prefix, a register id, a provider id and a subscriber id to form a valid provider-based global address. The *IPv4 compatible IPv6* address is targeted for the transition phase. This address will include the IPv4 address in the lower 32 bit of the IPv6 address prefixed by 96 zeros. This address allows automatic tunneling of IPv6 packets through IPv4 infrastructure. This feature will be described in more detail in Section 4.

Anycast address

An anycast address indentifies a set of different interfaces on one or more nodes. The packets will be delivered to the interface based on the routing protocols interpretation of the closest interface. All receiving nodes must be configured to recognize the anycast address, and the routers need a binding between the anycast address and the unicast addresses of the nodes in this particular anycast group.

If this address type is combined with the Routing extension header and an appropriate routing protocol, a load balancing scheme can be created. This would of course require that the cost function is updated when the load changes. One possible area would be load balancing on a WWW site with a number of servers with replicated file systems.

Another use of an anycast address is to include the access router of several different ISPs, and let the routing protocol decide which one to use. A smart routing protocol could route the packets via the ISP providing the cheapest traffic rates or the highest bandwidth.

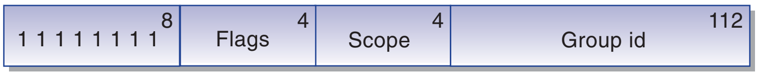


Figure 2 Multicast address

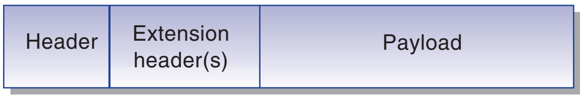


Figure 3 IPv6 packet format

Multicast address

The multicast address format, which is shown in Figure 2, will make it possible for a sender to transmit data to a predefined multicast group via a single multicast address. Multicasting was not available in the original IPv4 specification, but has been included later. One of the uses of this protocol add-on has been the MBONE which is a multicast network encompassing a large part of the Internet. The problem with the IPv4 multicast solution apart from the necessary protocol updates and patches to make it work is the scalability issue. In IPv6 this issue is solved with the introduction of a scope field. The 4 bit scope field is set to indicate the scope of the multicast group. Valid configurations are; node-local, link-local, site-local, organization-local, and global. By carefully using the scope field to indicate the scope of the multicast group, the multicast solution in IPv6 will scale significantly better than the add-on solution which exists for IPv4. The 4 bit flag field is used to indicate whether the multicast address is a permanently IETF assigned multicast address or is a temporary multicast address. The temporary multicast addresses are only valid within the assigned scope which makes it possible to reuse the address outside the corresponding scope. For permanent multicast addresses, the scope field limits the addressing scope of the IPv6 packets.

Multicast has several interesting uses apart from audio and video conferences. It can for example be used to emulate a broadcast option by letting all nodes, e.g. routers, which should get a control packet of some sort be included in a multicast group within an appropriate scope.

2.3 Extension headers

IPv6 has reduced the complexity of IP routing by moving all options out of the fixed-sized packet header. This will improve the packet per second routing capacity of IPv6 routers. This is important since the backbone capacity increases more rapidly than forwarding capacity of IP routers. The variable option field has been moved to extension headers which are inserted between the IPv6 header and the IPv6 payload as illustrated in Figure 3.

The IPv6 specification has defined six extension headers:

- Hop-by-Hop options header
- Destination options header[1]
- Routing header
- Fragment header
- Authentication
- Encapsulating Security Payload.

Each extension header includes a next header field which points to the next extension header if present. The size of an extension header is an integer number of 8 bytes to maintain the alignment which is crucial to protocol performance.

The ordering of these options which is outlined above is important for the efficiency of the IPv6 routing. The options which have to be examined by all routers, e.g. Hop-by-Hop options, are the first options after the IPv6 packet header. Thus, a router which forwards IPv6 packets will only need to examine the options which are relevant. In IPv4 a router needed to examine all the options.

An obvious advantage of the flexible mechanism for extension headers is that new functionality can be included without causing problems for other implementations. Unknown header extension headers should be discarded. Another advantage is that these headers are ordered so that options which are not relevant to a router will not be processed at all. This makes the routing of IPv6 packets very efficient.

Hop-by-Hop options header

The Hop-by-Hop options header includes information which should be processed not only in the sender and receiver but in all intermediate routers. This is the only extension header which is processed until the packet reaches the node indicated in the destination address. The only option type which has been specified so far is the jumbo payload type which allows the transmission of packets larger than

[1] *The Destination header can be placed as the last extension header and before the Routing header. This option should be processed both by the final destination and potentially by all intermediate routers specified in the Routing header. See the description of the Routing header for more information.*

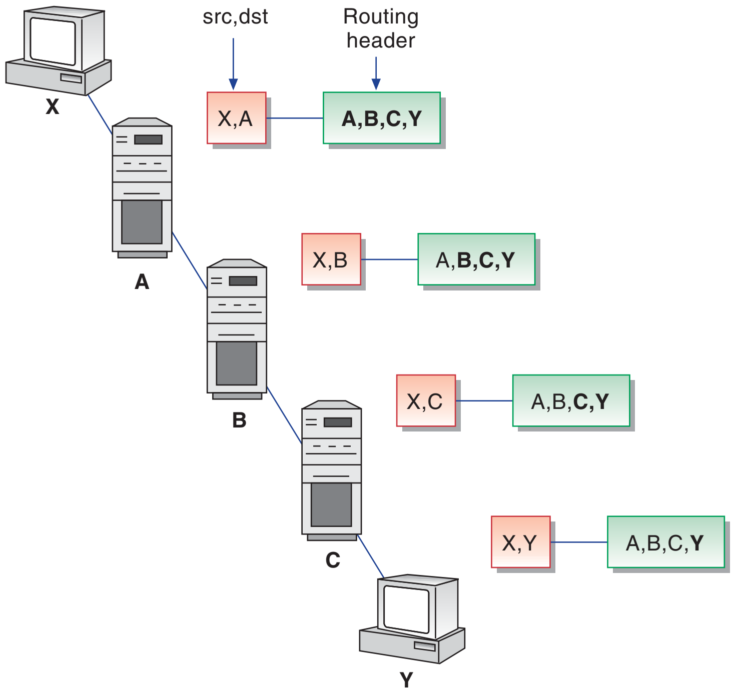


Figure 4 Use of Routing Header

64 kbyte. The 32 bit data field of this option will give the size of the large IPv6 packet excluding the IPv6 header. Other option types can of course be specified when they are needed.

Destination options header

The Destination option header is only processed by final destination identified by the IPv6 destination address. This will also be the case if a Destination option header prefixes a Routing header as illustrated in Figure 4. Here the Destination option header, if present, will be processed by all the intermediate routers in the Routing header.

Routing header

The Routing header makes it possible for a sender to decide on a set of intermediate routers an IPv6 packet should visit. This operation is illustrated in Figure 4, where end-system X is transmitting data to end-system Y with a valid Routing header. The sending end-system will set the destination address in the IPv6 header to the IPv6 address of the first intermediate router in the Routing header. In the example in Figure 4, when the packet arrives in node A, the router will shift the IPv6 address of the next intermediate router in the Routing header list into the IPv6 destination address in the IPv6 header. By shifting the next intermediate router into the destination address, the Routing header forces the packet to visit all the nodes A, B and C between the two communicating end-systems X and Y. The last address in the Routing header is the address of the final destination.

Fragment header

IPv6 does not allow packet fragmentation within the network for efficiency reasons. Only the sender is allowed to fragment a packet which must be smaller than 64 kbyte. Fragmentation will be necessary if one or more links have MTU sizes less than the packet size used by the IPv6 node. This option will not be used to a great extent since most implementations will employ Path MTU discovery [15]. Routers supporting Path MTU discovery will reply with an ICMPv6 [6] Packet-to-Big message to the sender if the next hop of the route does not allow the IPv6 packet to be forwarded. The ICMPv6 message will include a suggested MTU size which is allowed on the next hop. The IPv6 sender will continue to iterate through the network in order to find a path which can support a certain MTU size which is also acceptable to the sender.

Authentication header

The authentication header provides authentication and data integrity to the users. This extension header will be discussed in more detail in the IPv6 security subsection later in this article.

Encapsulating Security Payload header

The encapsulating security payload extension header provides privacy and confidentiality to the users. This extension header will be discussed in more detail in the IPv6 security subsection later in this article.

3 Internet protocol version 6 – functionality

3.1 Address autoconfiguration

Address autoconfiguration is one of the features which will be important in IPv6. It should not be necessary to manually configure individual end-systems before connecting them to the network. This is absolutely necessary for nomadic end-systems. IPv6 defines two different address autoconfiguration schemes denoted stateful and stateless address autoconfiguration.

The latter scheme does not require any manual configuration of hosts and only minimal configuration of routers. This scheme does not need any servers with address information. An end-system using this scheme will generate its own address based on local information and information (e.g. subnet prefix) which is distributed by subnet-routers. These advertisements can contain the link-layer prefix, the netmask and MTU information. This autoconfiguration scheme is useful for small and large sites where the configuration of an address server is too costly. It will also make it easier to change the existing number plan by letting the router distribute a new address prefix instead of renumbering each end-system manually.

A host using stateful autoconfiguration (e.g. DHCPv6 [16]) will obtain interface addresses and other configuration parameters from a server. These servers will maintain a database of all the hosts and their addresses. The stateful autoconfiguration will be used when the address assignment requires a tighter control than the stateless configuration scheme offers. The two schemes can of course be combined, e.g. where the stateless autoconfiguration finds the link-local address and the stateful autoconfiguration retrieves additional information from an address server.

Address autoconfiguration is closely linked to another necessary IPv6 feature denoted Neighbor Discovery [17]. This mechanism is used by IPv6 nodes to determine the link-layer addresses of other nodes on the same subnet and to quickly invalidate cache entries which become obsolete. Hosts also use this mechanism to find routers that are willing to forward packets on their behalf. Finally, the mechanism is used to keep track of which neighbors are reachable and potential changes of the subnet address prefix. Router reachability is very important of course, especially when nodes are moving between subnets.

3.2 Quality of Service

In the previous IP version, there was no support for any Quality of Service based internetworking. Due to the anticipated need for QoS based internetworking, IPv6 has introduced support for QoS parameters based on a flow label and a priority field in the IPv6 header.

Flow label

IPv6 includes a flow label which is used to provide special handling to certain flows in intermediate routers. This label can be used by end-systems to request similar handling of packets belonging to the same logical flow or packets from flows with similar characteristics. The flow label has no practical use unless the router has a binding between the flow label and the requested QoS of this flow. This information including necessary traffic descriptors will be distributed to the intermediate routers via a resource reservation protocol (e.g. Resource Reservation Protocol (RSVP) [19]) or in a new extension header. Which of these solutions which will be used will depend on the scalability of RSVP. Traffic flows which do not need special handling will have a zero flow label.

Intermediate routers which do not support special handling of any flows will not use the flow label information. Currently, the backbone capacity is increasing at a higher rate than the routing capacity. Several solutions, including IP switching and Tag switching, have been proposed to improve the routing performance. The IPv6 flow label can play an important part in these and other label switching solutions to provide useful information about flows which can benefit from switching compared to traditional forwarding.

Priority

The QoS feature in IPv6 includes a priority field which can be used in combination with the flow label to provide information to the intermediate routers. The 4-bit priority field is used to identify the relative priority between packets from a single source. Priority levels 0–7 are used to indicate the priority for connections where the higher layer protocol will apply a congestion control mechanism if congestion occurs. One example is the congestion control and avoidance algorithm used by TCP to back-off during situations with severe packet loss. Other protocols may use similar congestion control algorithms. Priority level 0 is used for Internet control traffic which is the most important traffic to get through the network during congestion. Other traffic classes in decreasing priority order are interactive traffic, attended data transfer, bulk data transfer, store-and-forward traffic and uncharacterized traffic.

The upper range of the priority levels (8–15) is used for traffic where the higher layer protocol has no congestion control mechanism which will reduce the traffic in case of congestions. This traffic class is ideal for audio and video traffic which have real-time characteristics and where retransmission is not an option. Priority level 8 is the lowest priority for the non-congestion controlled traffic, while 15 is the highest priority level.

The priority levels of the two sub-classes cannot be compared since these priority levels are independent. The priority fields only have meaning within each sub-class.

3.3 Routing

The routing in IPv6 is very similar to the CIDR-based routing in IPv4, and all the routing protocols (e.g. OSPF, RIP) can be reused with straightforward modifications due to the increased address size.

However, IPv6 includes some new features which makes it possible to move to other locations, to choose network provider based on a cost function, and automatic re-addressing.

This functionality is made possible by the Routing header which was described in sub-section 2.3 above. IPv6 requires that an end-system receiving traffic with a Routing header returns the traffic with a reversed Routing header. Thus, IPv6 can handle end-systems that change their point of access without breaking the higher layer protocol (e.g. TCP) connec-

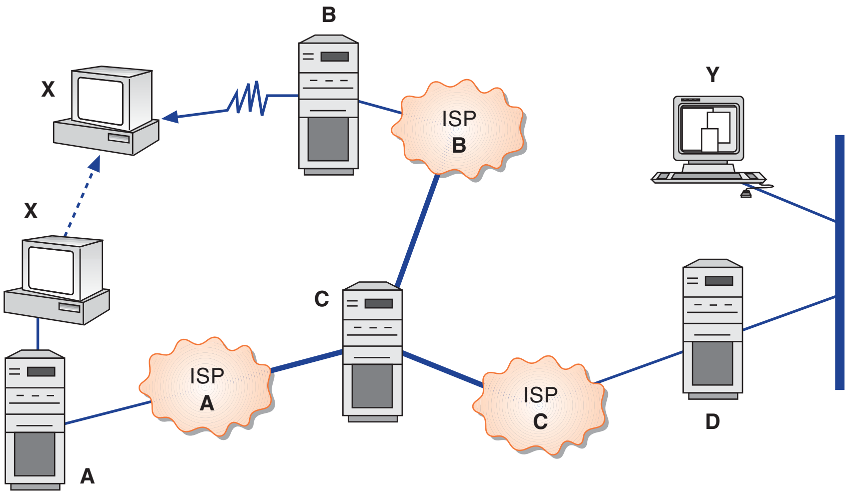


Figure 5 IPv6 routing features

tivity. An example is shown in Figure 5, where end-system X is moving from Internet Service Provider (ISP) A to ISP B. By changing the router list in the Routing header from A,C,D to B,C,D, the receiving end-system Y will start sending the IPv6 packets to the new location of end-system X. This solution makes it possible for end-system X to move without breaking any TCP connections to end-system Y.

3.4 Security

In Internet today there are no built-in security mechanisms. The existing security solutions are application and API specific (e.g. SHTTP, SSL, etc.). As opposed to IPv4, IPv6 has a built-in security mechanism providing both authentication and privacy to all higher layer protocols and services [4, 5].

Security associations

For secure communication, IPv6 needs to establish a unidirectional security association 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 lifetimes. Each association is unidirectional which means that a bi-directional connection needs one security association in each direction.

Authentication

The Authentication header offers both data integrity and authentication of IP packets. The Authentication header includes a length field, an SPI and the authentication data.

The standardized authentication algorithm which will be used within Internet is keyed MD5 [11]. This algorithm is calculated over the entire IP packet excluding protocol fields which are modified in intermediate routers, e.g. hop limit field. These fields are set to zero prior to the keyed MD5 calculation in both the sender and the receiver. Since the calculation is done prior to and after the fragmentation and reassembly processes, fragmentation of packets is possible.

Authentication can be done between the sender and the receiver or between the sender and a firewall. The latter scenario is useful if the firewall can authenticate on behalf of all the machines behind the firewall. The advantage of this scheme is that fewer authentication keys are necessary which results in less key administration.

Privacy

The Encapsulating Security Payload (ESP) provides data integrity and privacy to the users. The algorithm which has been standardized by IETF is the DES-Cipher Block Chaining algorithm. The ESP header starts with a length field and a 32 bit SPI. The rest of the header, if present, contains parameters which are dependent on the algorithm which is in use. Parts of the ESP header including the SPI are transmitted unencrypted. This privacy mechanism can be used in two ways depending on the requested 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 technique is denoted Tunnel Mode ESP. The difference between these two modes is illustrated in Figure 6.

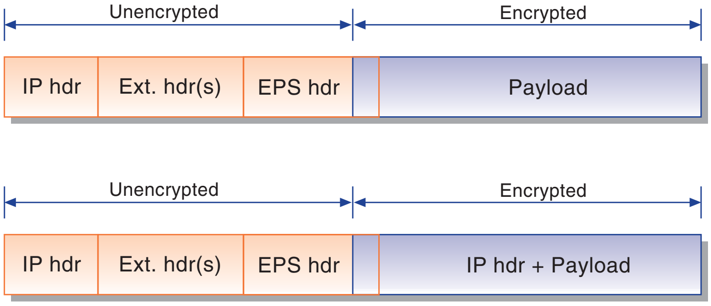


Figure 6 Transport (upper) and Tunnel (lower) mode ESP

Transport mode offers confidentiality to all upper layer protocols by introducing very little overhead. The drawback is that it is possible to do traffic analysis on the IP packet due to the fact that the packet is addressed to the final destination.

Tunnel mode ESP has more overhead than the Transport Mode but it prohibits traffic analysis by encrypting the entire IP packet. This mode is ideal for communication between nodes on both sides of a firewall. The firewall and the external system can use Tunnel Mode ESP to provide encryption for all internal systems.

3.5 Mobility

One of the features which will be important in the next generation Internet is mobility. The example shown in Figure 5 illustrates that an IPv6 node can move between subnets without even breaking existing TCP connections. However, a more sophisticated mobility solution is needed if other nodes should be able to contact a mobile node located outside its home subnet.

Figure 7 presents an overview of the mobility architecture[2] of IPv6 [18] where mobile node A communicates with a peer node denoted correspondent node.

The mobile needs a home agent (HA) in its home subnet which can forward packets when the mobile node is not connected to its home subnet. In addition, the mobile node needs a Care-of-address (COA) in the Foreign subnet it is connected to. The binding between the Home address (i.e. A) and the current Care-of-address (i.e. FA) will be maintained in a table in the Home agent.

The Mobile Node will always be addressable via the Home Agent which contains the primary Care-of-address of the mobile node in its binding table. When the Mobile node is connected to its Home subnet, normal IPv6 routing is used. The Home agent is using a proxy Neighbor Discovery to snoop on packets which are destined for a Mobile node with a valid Care-of-address binding. All packets with valid bindings will be tunneled by the Home agent to the Care-of-address of the Mobile Node.

The Care-of-address will be found probably by the stateless address autoconfiguration scheme in the Mobile node when it connects to the Foreign subnet. After the Mobile node has received a

[2] *IPv6 mobility is still an Internet draft.*

new Care-of-address, this address is distributed to the Home agent via an IP packet with an IP Destination Header. Currently, three option types (Binding Update, Binding Acknowledgement, and Binding request) have been defined. The Mobile node will send a Binding Update to the Home agent with the new Care-of-address.

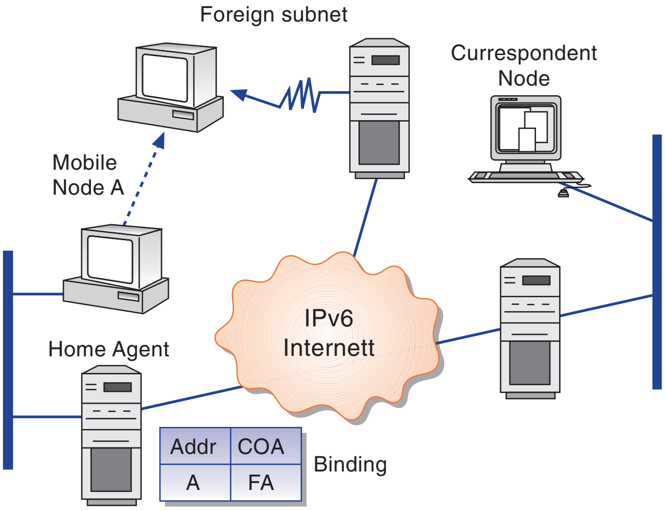


Figure 7 Mobile architecture

The routing from the Mobile Node to the Correspondent Node will be normal. Routing in the opposite direction will initially go via the Home agent. When the Mobile node receives the IP packet, it will realize that the correspondent node does not know the Care-of-address. In that case, it will distribute the address to the correspondent node. After the Correspondent Node has received the binding, the routing will bypass the Home agent and go directly between the two nodes by using Routing headers. The correspondent node will cache the binding and use until it times out. Prior to this time-out, it will send a Binding request to the Home Agent to get an updated binding. The rate of Binding Updates are limited to one per second.

An important part of the mobility architecture is the capability of the mobile node to detect movement. This detection is done via router discovery and Neighbor Unreachability detection of the Neighbor Discovery mechanism. A mobile node will detect that it is in the Home subnet by detecting that the network prefix is the same as the Home address.

The mobile solution outlined for IPv6 also contains support for multicasting. When a mobile node is in the Home subnet multicast support is of course trivial. However, multicasting is possible even when the mobile node is connected to another subnet.

A mobile node outside its home subnet can join a multicast group by contacting a multicast router in the Foreign subnet by advertising the Care-of-address in the multicast group membership control messages. Another solution is to make a tunnel to the Multicast router in the home subnet, i.e. Home agent. Sending packets to a multicast group can also be done in two ways. The first method is to send packets directly on the Foreign subnet by using the Care-of-address as the IPv6 Source address. The source address is used by the multicast routing. The other solution is to send the multicast packets via the Home agent. In this case the Home address is used as IPv6 Source address both in the multicast packet and the tunneling packet.

4 Transition mechanisms

The most important requirement of the IPv6 development process was to ensure a smooth upgrade path from the existing Internet Protocol. History has shown that similar technological advances must provide backward compatibility to become a success. It is implausible to upgrade the Internet or even larger parts at once. Thus, the two technologies will probably be used concurrently in different parts of the Internet for years to come.

The transition mechanisms [8] include use of complete protocol stacks for both IPv4 and IPv6 in hosts and routers, and tunneling of IPv6 over IPv4 routing infrastructures. These mechanisms should make an incremental deployment of IPv6 simpler and a final transition possible. The only requirement to deploy IPv6 is an upgrade of Domain Name Server (DNS) [7].

Domain Name Server (DNS)

Since the address format has been changed, the DNS system must be upgraded to handle both the IPv4 and the 128 bit IPv6 addresses simultaneously.

A new resource record denoted “AAAA” is defined for IPv6 addresses. Dual capable IPv4/IPv6 nodes must have resolver libraries which can handle both the IPv4 and IPv6 record types. For IPv4 compatible IPv6 addresses both the IPv4 address and the compatible address are registered in the “A” and “AAAA” resource records in DNS. The generated traffic (i.e. IPv4 or IPv6) will depend on the record type of the DNS query response.

Dual protocol stacks

The easiest way to guarantee interoperability with both IPv4 and IPv6 nodes is to use dual protocol stacks supporting both IP versions. A dual capable IP node can have separate IPv4 and IPv6 addresses or use an IPv4 compatible IPv6 address. The latter address format prefixes the 32 bit IPv4 address with 96 zeros to form a legal IPv6 address. These nodes can use the stateless IPv6 address configuration or DHCP to acquire an IPv6 address. IPv4 addresses can be found via standard IPv4 mechanisms (e.g. DHCP, BOOTP, RARP, manual configuration). IPv4 address can be mapped into an IPv4 compatible IPv6 address by prepending the zeros. This operation can be useful in networks where IPv6 address configuration services are not yet in place.

Tunneling

There are two different tunneling schemes which will be used by IPv6. Configured tunneling of IPv6 over IPv4 where the IPv4 address of the tunnel endpoint is determined by configuration information in the encapsulating host or router. In Automatic tunneling, the address of the IPv4 tunnel endpoint is determined by the IPv4 address embedded in the IPv4 compatible IPv6 address.

In either of the tunneling solutions, each IPv6 packet is encapsulated in a IPv4 packet and transmitted between two IPv4 tunnel endpoints. Tunneling can be used between all four combinations of host and router endpoints.

When IPv6 packets are tunneled from a host or router to an intermediate router, the tunnel endpoint is not the final endpoint. This type of tunneling is denoted configured tunneling since the node encapsulating the IPv6 packet needs some additional configuration to find the tunnel endpoint. The address of the tunnel endpoint can be an anycast address if several routers want to carry the traffic to the final destination. In that case, the encapsulating node will use the closest router to forward the packet closer to the destination.

When the final endpoint is the destination host which has an IPv4 compatible IPv6 address, the tunnel endpoint address can easily be found. This type of tunneling is denoted automatic tunneling since no configuration is needed in the encapsulating node.

There are some issues which must be solved by the nodes on both sides of the tunnel. ICMPv4 error messages must be converted to comparable ICMPv6 messages to inform the IPv6 node about failure situations. In addition, the nodes must handle the possible MTU size in the IPv4 tunnel and the IPv6 network encompassing the tunnel, and possibly fragment the IPv6 packet over the IPv4 tunnel.

5 Concluding remarks

IPv6 will be the next dominating Internet Protocol. The reason is not only the extension of the addressing space but also that it has a set of new features (e.g. security, mobility, multicast, routing functionality) which will make new applications and extended Internet services possible. Combined with the fact that operation during the transitional phase is an integral part of the IPv6 protocol specification, IPv6 will replace IPv4. At the moment, most of the computer and router vendors and important software companies delivering Internet protocol suite products are implementing IPv6. The Internet community has also launched a test network for IPv6 called 6BONE [20]. The purpose of this network is to gain experience of running large IPv6 network and interoperability tests between different protocol implementations. The results from these and other tests will be crucial in the future full scale deployment of IPv6.

References

1 Postel, J. Internet Protocol version 4. *Internet Engineering Task Force,* RFC 791.

2 Deering, S, Hinden, R. Internet Protocol, version 6 (IPv6) Specification. *Internet Engineering Task Force,* RFC 1883.

3 Hinden, R, Deering, S. IP version 6 Addressing Architecture. *Internet Engineering Task Force,* RFC 1884.

4 Atkinson, R. IP Authentication Header. *Internet Engineering Task Force,* RFC 1826.

5 Atkinson, R. IP Encapsulating Security Payload. *Internet Engineering Task Force,* RFC 1827.

6 Conta, A, Deering, S. Internet Control Message protocol (ICMPv6) for the Internet Protocol version 6 (IPv6) specification. *Internet Engineering Task Force,* RFC 1885.

7 Thomson, S, Huitema, C. DNS extensions to support IP version 6. *Internet Engineering Task Force,* RFC 1886.

8 Gilligan, R, Nordmark, E. Transition mechanisms for IPv6 hosts and routers. *Internet Engineering Task Force,* RFC 1993.

9 Hinden, R M. IP next generation overview. *Connexions,* 9, (3), 2–18, 1995.

10 Stallings, W. IPv6 : the new Internet Protocol. *IEEE Communications,* 34, (7), 96–108, 1996.

11 Metzger, P, Simpson, W. IP authentication using keyed MD5. *Internet Engineering Task Force,* RFC 1828.

12 Rekhter, Y, Li, T. An architecture for IP address allocation with CIDR. *Internet Engineering Task Force,* RFC 1518.

13 Huitema, C. The H ratio for address assignment efficiency. *Internet Engineering Task Force,* RFC 1715.

14 Stallings, W, Wright, G R. *TCP/IP Illustrated, vol. 2.* Reading, Mass., Addison-Wesley, 1995.

15 McCann, J et al. Path MTU discovery for IP version 6. *Internet Engineering Task Force,* RFC 1981.

16 Droms, R. Dynamic Host Configuration Protocol. *Internet Engineering Task Force,* RFC 1541.

17 Narten, T et al. Neighbor Discovery for IP version 6 (IPv6). *Internet Engineering Task Force,* RFC 1970.

18 Johnson, D B, Perkins, C. Mobility support in IPv6. *Internet Engineering Task Force,* Internet draft (work in process).

19 ReSerVation Protocol home page, *http://www.isi.edu/div7/rsvp/*

20 6BONE home page, *http://www-cnr.lbl.gov/6bone/*

21 Lefler, S J et al. *The design and implementation of the 4.3 BSD UNIX Operating System.* Reading, Mass., Addison-Wesley, 1989.

Espen Klovning is Research Scientist at Telenor R&D with interests covering all aspects of high performance communication systems. He is currently a member of the Internet unit within Telenor R&D and works with the next generation Internet technology.

email address:
espen.klovning@fou.telenor.no

Aquarius: A web-based multimedia information service

KUROSH BOZORGEBRAHIMI AND ERIK DAHL

Advances in the development of server technology and related software allows relatively easy installation of video storage and retrieval systems in intranet/Internet environments. Telenor has built a service based on such equipment in Trondheim for the EU Fourth Framework project Aquarius which aims at a sustainable electronic education and information service for the aquaculture sector. The CATV network is being used for accesses.

1 Background

Considerable attention has recently been given to the integration of video and multimedia in World Wide Web and intranet applications. In education this opens up for many new techniques. Video clips can be used for short sequences during class, multimedia applications are suitable for computer-based training from home, etc. Such video clips can also be used by entrepreneurs for on-the-job training and to optimize production processes.

The aim of the Aquarius project is to develop *a sustainable electronic education and information service* in the aquaculture sector. The project is supported by the European Union under the Fourth Framework programme. This service shall be based on the integration of existing and emerging networks, telematic technologies and protocols. By doing this one facilitates flexible communication, information exchange and collaborative work between academic research centres and businesses. In Norway the latter includes fish farms along the coast. Participants in the project are the universities of Wageningen, Ghent, and Trondheim. Telenor is one of several sponsoring partners of the Aquarius project.

Telenor's main contribution is to develop and operate a telecommunication service based on a video server. Our responsibility is to supply hardware and software, lines etc. to establish the service. In addition, Telenor is responsible for operation of the service and support up to an API interface. There are two phases of the implementation. The first aim is to establish such a service locally to selected users in Trondheim. Secondly, a similar service will be made available to a broader public – although the quality of the video will necessarily be lower in this case than in the local service.

The aquaculture sector is a rather young business; still there are approximately 20,000 people engaged in it in Norway. In Belgium and the Netherlands, two other countries with participants in the Aquarius project, the number of fish farms is much smaller: less than one hundred in each country.

Telenor's efforts have been led from our offices at Tyholt, Trondheim. Much of the work, including implementation and operation of the media server, has been done by Telenor Nett while Telenor Avidi have assisted with the CATV distribution system and Telenor Bedrift with supplementary services. Since the project is a close co-operation between NTNU and Telenor, video signals are available on the Campus network as well as the CATV network, and Internet connection is accomplished through a University node.

2 Functionality of the video-streaming service

2.1 Selection of technology

The Aquarius contract specified a video-on-demand service allowing ISDN and IP based retrieval. In a separate subcontract between NTNU and Telenor the latter was given the task to implement a service based on a video server. NTNU has the responsibility for content as well as organization of data, navigation, Intellecutal Property Rights, etc.

The contract mentions the upgrade of an information server to a video server as an objective. Telenor has chosen to keep the two separate for obvious reasons. A video server is far more advanced and needs another architecture and other optimizations than a general web server. On the other hand, the video (or media) server chosen could easily be used as a general web server as well as a pump for audio-visual information.

To comply with the Aquarius contract Telenor supplied a general web server to NTNU's facilities for installation of the ICE database, etc. The media server, however, was installed at Telenor's premises.

The media server for the project had to meet many requirements. It is stated in Work Packages 5-6 of the Aquarius contract that a service should be available on both ISDN and IP although this is somewhat ambiguous as IP can be supplied over ISDN as well as other networks. We chose to interpret it to mean that the video signal should be available on various formats including MPEG1 and H.263, making it possible to see videos on POTS, ISDN, CATV, ADSL, Ethernet, and ATM networks. In addition, there were requirements to flexibility, compatibility, reliability, and not least, a reasonable price.

Several vendors were considered for the media server and related functions. An early candidate was Video Call from Telsis which seemed well suited for ISDN, but it could not readily deliver the desired MPEG functionality. DEC equipment was used for a similar project conducted by Telenor Research in Oslo (the VideoTorg project), but their system was not suited for the web environment and it was rather expensive.

A report published in *Network Computing* 15 October 1996 with the title "Video servers: Live from your network" compared high-end video servers from four vendors: First Virtual Corporation, Starlite Networks, Oracle, and Silicon Graphics (SGI). The conclusion was clear: SGI's equipment was by far the most feature-laden, complete, and easy-to-install solution in the market at the time. In addition, it turned out a fairly reasonable solution.

2.2 System installation

Figure 2.1 shows a sketch of the SGI media server installation which was completed at Telenor's premises at Tyholt.

The network consists of the following components:

- Challenge S media server with disc cabinet
- An SGI INDY work station with software necessary to create content and manage it
- An IBM Aptiva PC with MPEG1 codec card. The PC is also connected to a video player and a TV set. Analogue video can be encoded in MPEG1 format and loaded to the server
- The HW video encoder supplied by Optivision.

The installation of the server etc. went straight-forward, with good help from SGI personnel. There were some

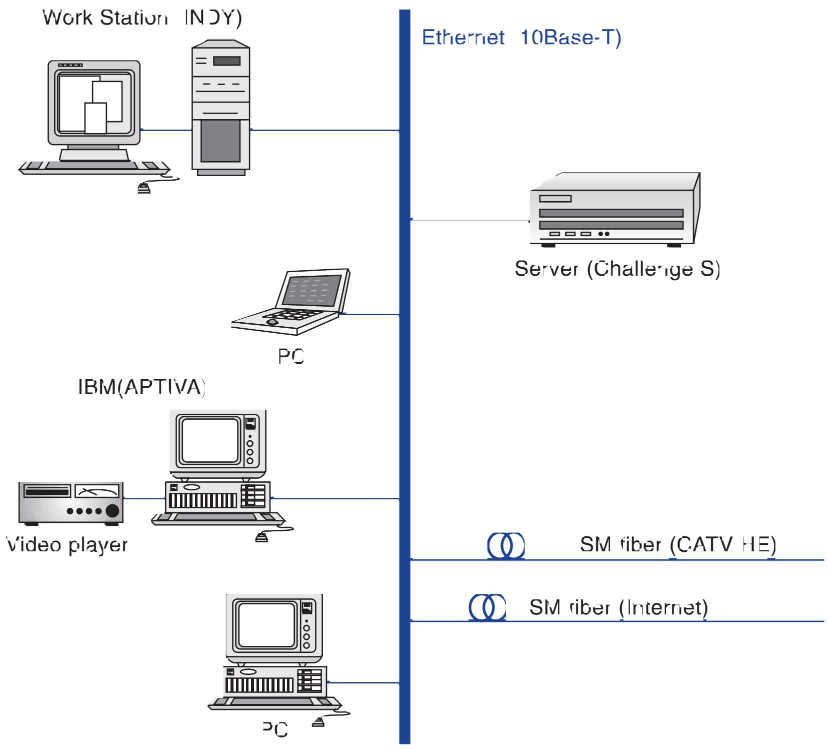


Figure 2.1 Media server installation for the Aquarius project at Telenor's premises

problems with the Ethernet connection between the Telenor building and the Tyholt tower (the CATV head-end) due to mismatch in signal levels and unavailability of multimode/single mode conversion units. These problems were eventually solved.

The initial installation (November 1996) contained MediaBase release 1.0 with limited functionality. Only MPEG1 encoding was available at that stage. Later, by June 1997, version 2.0 of the software was implemented in the server, allowing a series of new functions including H.263 with bitrates from 28.8 kb/s (POTS). ISDN can be used with bitrates 64 kb/s and 128 kb/s, and MPEG2 can be coded up to 8 Mb/s. Other new features of version 2.0 include improved content management, RSVP support, live video capture, bitrate-on-demand functionality, support for VXtreme and Progressive Network encoding, etc.

Several videos of various formats were loaded into the media server and can be seen from a client connected to the system provided sufficient bandwidth of the line. The client could be a workstation or a Pentium PC with minimum 133 MHz clock frequency, at least 16 MB RAM and 256 kB cache memory, Windows 95 or NT, Netscape (2.0 or above) and MPEG1 (or H.263) player software. The latter can be downloaded from the network, see Section 2.4.

2.3 Discussion of testing results

As mentioned above it was decided to install a separate server for video storage and retrieval, not to upgrade the general web server to a video (media) server as prescribed in the Aquarius main contract. The testing of the media server has consisted in retrieval of video information by Telenor and the various users.

The server (with WebFORCE MediaBase, see details in the Enclosure) can display videos in two formats: the Plug-in and the Helper. The former was rather reliable in most cases while the Helper failed sometimes. As version 2.0 of the software was installed by June 1997 the situation improved considerably.

2.4 Description of the technical demonstrator

The server and its related equipment (Section 2.1) is connected to the Internet through NTNU's node according to a special agreement between NTNU and Telenor. The web-address is *http://mediabase.itea.ntnu.no/mbase*. It can also be reached through the Aquarius homepage *http://no.aquarius.euro.org*.

As of 1 July 1997 altogether 57 video clips of duration 2 hours and 37 minutes (combined) were available on the server, most of them in Norwegian language. However, the process of loading English-spoken video clips (and other languages) was underway. There is presently disc space for storage of approximately 13 hours video (in MPEG1 format). This capacity can be increased on demand.

To see the video clips users must download the decoder software (and as mentioned before have a PC with sufficient speed to handle video, and an appropriate line). The necessary software is available on the media server homepage (*http://mediabase.itea.ntnu.no/mbase)* or on the Internet. If you enter through the homepage, choose "Install MediaBase Video Players". In addition to MPEG1 and H.263 videos it should be noted that MediaBase supports other useful formats like RealAudio/RealVideo form Progressive Networks.

3 The CATV access network

3.1 Choice of access technology

There is a need to connect the users to the Aquarius service. Several optional access technologies are available: Plain Old Telephone Lines (POTS), ISDN, ADSL (or HDSL, SDSL) modems, CATV lines with modems, radio links, satellite transmissions, etc. Assuming there is a need to transmit video signals with a bandwidth exceeding 1 Mb/s, POTS and ISDN are ruled out, and radio transmission and satellites are either impractical or too expensive. The "hottest" candidates in this case are ADSL and CATV access with modems. Of course, in many cases of Aquarius video is not required. Then all the available access technologies mentioned can be used.

It was decided to try out CATV access in Trondheim in the Aquarius project. The reasons for this included availability of an extensive CATV network in that town, appropriate technology, and a will on the behalf of the operator to test it. Such access networks had not been realized in Norway before. Of course Telenor is testing other technologies as well, including ADSL, but such tests are carried out in other projects.

Several vendors were considered as suppliers of cable modems and related equipment. A rather thorough selection process was carried out by Telenor Avidi, the audio-visual subsidiary of the Norwegian telecom operator. LAN City (owned by Bay Networks) was chosen because of its documented reliability, flexibility, and reasonable cost. An extensive description of the modems and the related management system is given in Bay Networks' web-site *http://www.baynetworks.com*.

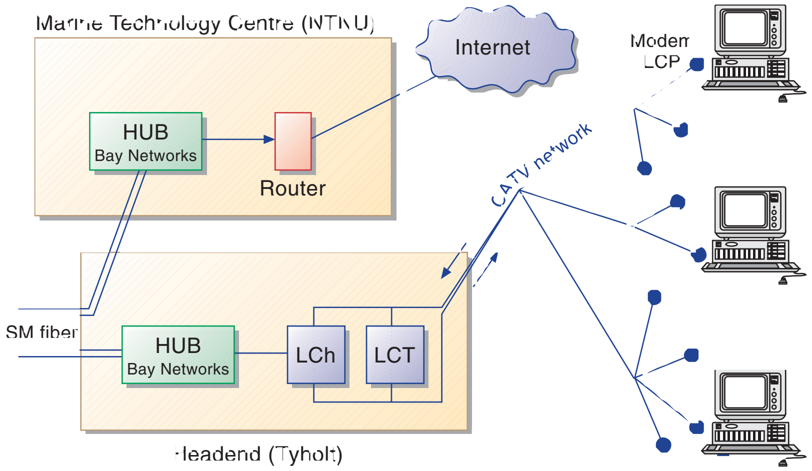


Figure 3.1 CATV access network realised by Aquarius in Trondheim (principle sketch)

3.2 Network architecture

Figure 3.1 shows the main components of the CATV distribution system built in Trondheim. Three vital components are seen in the figure: LCh, LCT, and LCP.

LCh is the main modem at the CATV head-end. It is the reference point for the other modems. Each LCh supports one channels. LCT is the frequency converter at the head-end. LCP (Personal Cable modem) is the smallest of LAN City's user modems and it is capable of communicating with one PC only. The modem stores the IP address of the PC for later use, and if the address is changed, the modem must be reset.

In addition to the above-mentioned components there was installed a LCn unit: a management station to supervise the CATV access network and remotely control the modems. It is possible from the LCn to keep track of the history of usage and failures of each modem in the system.

Several problems were encountered during the establishment of the system. Much of this is due to the fact that it was a number one installation in Norway. Once in place the system exhibited a satisfactory record of operation.

Problems encountered in the establishment phase included:

- The CATV network is built for TV transmission in one direction only and a modification is needed to allow for two-way communication.
- The network has been built throughout a fifteen-year period and various generations of amplifiers etc. are used. Only the newest part of the network could readily be upgraded.
- The work was complicated by the fact that available expertise on CATV digital technology was scarce within Telenor. This was a number one installation and the technology was unfamiliar. However, there was a steep learning curve in the project phase.
- On short distances from the head-end (1–2 km), the available coax cables could be used. For more remote connections, separate fibre cables must be installed to guarantee a proper signal level to the remote end. The laying of fibre cables was delayed due to heavy snowfall, difficult cable routes, etc.
- A few users were not yet connected at the time of writing this report due to unforeseen difficulties in the fibre lay process.

The capacity of the network is 10 Mb/s in each direction to the users, but several users must share an access like in an ordinary Ethernet.

3.3 Discussion of testing results

The objective according to Deliverable 6.1 of Aquarius was to connect students houses (and possibly SMEs) via the CATV network. During the work to accomplish that end much experience was gained. It is impossible to include all minor problems encountered, but some main factors should be mentioned:

- It is important to keep track of (and control) the signal level into the houses and *indoors* as well. Each plug in the house ought to be monitored.
- There were some unexplained breaks in the connection, especially to remote users relying solely on the coax cables. Fibre-coax connections were generally more reliable. A study is recommended of BER as a function of noise and carrier frequency, short and long reflections on the line, the impact of radiation from household appliances and mobile phones, etc.
- Users experienced some inconvenience during the establishment phase and the initial usage phase. Once in proper service, the system was reliable and usage was unproblematic.

Telenor will continue to monitor the connections and experience will be collected throughout the remaining phases of the Aquarius project.

3.4 Operational experience

The Aquarius CATV access network was shown in principle in Figure 3.1. A fibre connection was realised from Telenor's facilities at Tyholt where the media server is located (see Section 2) to the CATV head-end at the Tyholt tower. Similarly, a fibre connected the Tyholt tower to the NTNU campus network at Marinteknisk senter. Ethernet (10 Mb/s) was used in these systems, but an upgrade to 100 Mb/s could be made readily on demand.

As of June 1997 nine users were connected through the CATV network with separate modems. They reported on a monthly basis on their usage of the system, the usefulness of the connection from their side, problems encountered, proposals for improvement of the system, and other related topics. Generally speaking, the comments are very positive. The speed of the Internet access is excellent (although one must foresee a somewhat slower connection if a large number of subscribers were to share an Ethernet segment). Videos, as described in Section 4, can be seen without difficulties. There have been few operational problems, most of which are due to general power failure (lightning, etc.) or finger trouble with the PCs.

All usage of the system and operational problems are recorded at Telenor's management station at Tyholt. The log shows that once in place there are few problems. However, we still experience some failures at a few locations, notably the ones with old CATV equipment and long coax connections.

4 Supply and encoding of video material

A major issue for video transfer on the web is Intellectual Property Rights. In a test situation it is easy for content vendors to supply video clips etc. for storage on a server and distribution to a limited number of test users. This was done in the initial phases of the Aquarius project. Negotiations are underway to handle the situation when the service becomes more widely available.

In fact, it is remarkable that the fish farming industry and related institutions have produced much video material of high quality. There has generally been a sound economy in the business, and many of the videos produced aim at selling Norway as a fishing nation. Consequently, many films display scenic nature of our long coastline.

Encoding in the MPEG1 format is accomplished through a hardware unit supplied by Optivision. A live feed unit will be purchased to directly transmit video in MPEG1. H.263 and RealAudio/ RealVideo encoding is carried out through software.

5 Perspectives for further usage

The Aquarius project runs until 1 January 1998 but an application has been made to the European Commission to prolong it with one more year. Although there was a difficult start, the project as a whole now exhibits considerable success. Telenor's multimedia service is just one of the elements – another is a very sophisticated database (named ICE) with object-oriented storage, SGML format, hierarchical categorization of content, etc. Among the strengths of the Aquarius project is the intimate linking to the users, both the academic and professional institutions in biology/fish farming and the local fish farms.

The access to video information on Telenor's media server via POTS and ISDN is in the initial phase and there is a considerable potential to be exploited. Since the academic institutions have access through an ATM network from Holland, it has recently been demonstrated that even MPEG1 coded video at 1.5 Mb/s can be seen from Wageningen!

Telenor Nett has already used the experiences from Aquarius in a similar project – *KontaktNett* – supplying four schools in Trondheim with broadband access to a media server and the Internet. In this case an Origin 200 server is used. The installation is operational and the pupils are collecting video material which is loaded into the server and used by classes collectively and by boys and girls for exercises. Given the ease of installation and relatively low cost, there should be a grand potential for similar installations by Telenor in many contexts in the years to come.

Kurosh Bozorgebrahimi is Senior Engineer with Telenor Networks, Division Infrastructure. He was employed by Telenor two years ago after graduating from the Norwegian University of Science and Technology. He is responsible for the implementation and operation of several pilot projects involving the use of media servers and broadband access technology.

email address:
kurosh.bozorgebrahimi@trondheim.nett.telenor.no

Erik Dahl graduated from the Norwegian University of Science and Technology in 1973 and completed his Dr.Ing. thesis in 1978. He joined Televerket (Telenor) in 1986 after previous employment in other companies. He has contributed to network planning and strategic customer projects, especially related to broadband networks and multimedia. He is Senior Engineer and manager of Network services within the Transport network division, Telenor Networks.

email address:
erik.dahl@trondheim.nett.telenor.no

Performance issues of Internet end systems

KJERSTI MOLDEKLEV

This paper exemplifies the performance issues of Internet end systems through experimental per-channel performance measurements within a local area network environment. The variation in measured performance dependent on actual values of configuration parameters is explained as the *end system behavior*.

1 Introduction

The utilization of the underlying physical network transmission capacity depends on the end system capacity, the network service, the network topology and routing, the end-to-end communication protocols, as well as the implementation of the distributed application. Already in 1981 the question of where to put the communication subsystem functions for optimal performance in distributed systems was touched upon [1]. Saltzer et al's end-to-end argument concludes that unnecessary and inefficient protocol processing is a result of not leaving the choice of service to the application. Instead, functions placed at low system levels may be redundant and of little value compared to the cost of providing them. From the end system point of view, the above end-to-end argument has a twofold interpretation. Firstly, the network service should be simple without excessive functions. Secondly, the higher level communication services should be flexible and configurable dependent on application requirements.

At the time of introducing a local area network technology, the network capacity is rarely the performance bottleneck. Meanwhile, the evolution of end system technology has made the network the bottleneck of previous generations of networks. In addition to low delay through the protocol stack, the *present end system challenge* is *efficient utilization of the transmission capacity of the high-speed networks*. To meet the end system challenge, the implementation structure and architecture of the protocol tend to be even more important than the protocol itself.

1.1 Directions of end system high-speed networking

End system communication speed may be increased by optimizing the protocol processing to perform fewer operations and/or by assigning more resources to the protocol processing [2]. This maps to the software- and hardware-oriented end system protocol directions of Figure 1, respectively. None of the directions represent *the* final solution to protocol processing in real transmission time. Combinations of the discussed techniques are, and will be implemented [3].

The *protocol specification* approach to high-speed end-to-end networking concentrates on the end-to-end transport protocols. Supporters of improving existing protocols argue that the source of protocol processing overhead is not the protocol itself, but its environment[1]. Improvements are mainly focused on the Internet TCP protocol. Supporters of designing new protocols claim that starting from scratch is necessary to take the new high-speed network environment in mind.

The *protocol processing structure* approach considers the whole protocol stack and the relation between its protocols and their functions. Generally, current protocol stacks have a layered specification and a layered processing structure. Clark and Tennenhouse emphasize the difference between protocol specification and protocol engineering [4]; to reduce data manipulation overhead, current and new protocols should use a non-layered structure which integrates the processing of different layers. Another protocol structure suggests to horizontally divide the protocols into functions which are invoked dependent on application needs and which can be executed without knowing the results of each other [5].

The *protocol implementation architecture* approach considers the integration of the protocol execution into the end system environment. In monolithic kernels the protocol processing is performed within the kernel. In micro kernels the protocols are usually implemented in servers outside the kernel itself. User level protocol processing may also be performed as part of the application through linkable library calls.

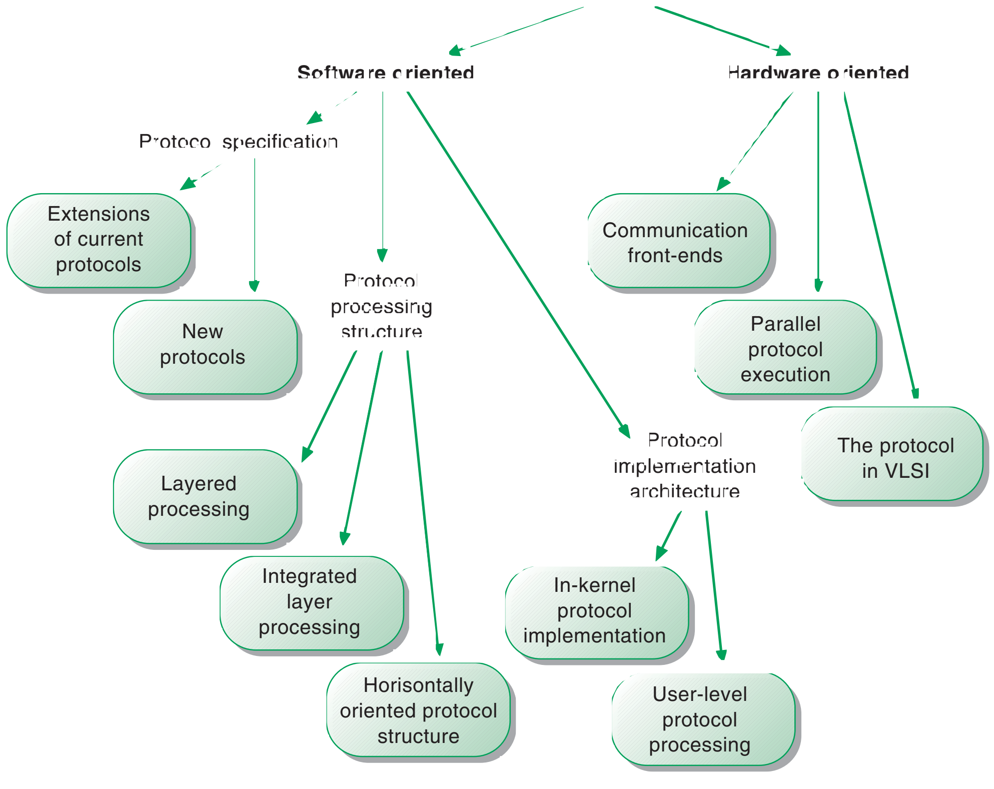

Software oriented
Hardware oriented
Protocol specification
Extensions of current protocols
New protocols
Protocol processing structure
Layered processing
Integrated layer processing
Horisontally oriented protocol structure
Protocol implementation architecture
In-kernel protocol implementation
User-level protocol processing
Communication front-ends
Parallel protocol execution
The protocol in VLSI

Figure 1 High-speed end system networking directions

[1] *The "20/80-rule" for end system communication claims that only about 20 % of processing involved is protocol-specific code.*

The *hardware-oriented* approach focuses on the use of additional processing resources to improve the protocol performance: Communication front-ends relieves the host of the protocol processing, parallel execution of protocols is based on multiprocessors, while the protocol in VLSI applies protocol-specific hardware.

The remainder of this paper focuses on performance issues and bottlenecks related to layered processing of in-kernel protocol implementation of the Internet TCP/IP protocols.

2 Experimental performance measurements

In this paper the quantitative method of evaluating the end system's networking performance and behavior is *performance measurements* [6] of the Internet TCP/IP protocol stack running on end systems attached to an ATM network. The advantages of using measurements to evaluate the protocol performance instead of simulation or analytic modelling are:

- The actual composite system including the protocol and its environment is examined and evaluated. The evaluation is based neither on a simplified system model nor single components.
- Due to the numerous parameters of the genuine system, any modelling or simulation of the protocol software and hardware environment is a non-trivial task. Furthermore, interactions which affect performance, and which are difficult to capture in a model, may be present.

The disadvantage of experimental performance measurements are:

- The need for a running system, partly dedicated to be able to perform controlled measurements.
- To study modifications of the end system software on the measured performance, source code licences of the operating system and network driver are required.
- Modifications to the computer hardware is impossible, and building a network interface is a highly resource-demanding task.

2.1 End system modules

Due to the numerous parameters affecting networking performance, the evaluation of protocol performance is comprehensive and challenging. Studying the protocol in isolation does not give all the answers as long as the evaluation is to include more than a single protocol mechanism. In addition to knowledge of protocol behavior, focus on the end system hardware and software implementation environment, and its interaction with the protocol implementation is required. A host upgrade does not result in a network performance increase proportional to e.g. the MIPS[2]-rating of the computer. This rating does neither represent networking performance, nor the limitations which are set by the protocol parameters and the bus transfers of data between the CPU and the main memory. This paper examplifies performance impact of host hardware and of the following modules:

- The *application* and its *parameters*
  The role of the benchmark application is to generate the measurement workload. Performance issues related to the application are the type of workload, and how the API is used to present the workload to the underlying communication subsystem; including both the size and memory alignment of the user data buffer.
- The *TCP protocol*, its *parameters* and *implementation*
  The connection oriented TCP and the connectionless UDP are the end-to-end protocols of the Internet protocol stack. UDP is a much simpler protocol than TCP, as TCP provides a reliable service; data is received free of errors, in order and without duplicates. TCP is chosen at the transport protocol level as the sending end system is capable of generating UDP packets faster than the receiving end system can process. A protocol including an end-to-end flow control is therefore needed to properly include the receiving end system in the measurements. In addition to the TCP window flow control, protocol mechanisms affecting measured performance are among others the acknowledgment scheme, the segmentation of the byte stream including among others the blocking of small segments as in Nagle's algorithm [7], and the action sequence on acknowledgment reception.
- Network *memory management* of the operating system
  The representation of packet header and data depends on the network memory management of the operating system. Both its data structures and functions contribute to the segmentation of the application data stream.
- The *network interface*
  Transmission and reception of network packets are the responsibility of the network interface. Its integration into the host architecture, and the network adapter functions above the physical layer are essential to achievable performance.

Both the per-channel performance and the performance of multiplexing several channels with different service quality requirements could be focused in the evaluation of end system performance. A comprehensive understanding of the per-channel performance is essential to the evaluation of the performance when several channels are multiplexed.

2.2 Measured performance

Generally, end-to-end protocol performance is measured in terms of time to set-up and close a connection, data transfer capacity or throughput, and response time which is the elapsed time between sending a request and receiving its response [8]. Other end system measures are CPU processor load and instruction counts [9]. This paper focuses on the *throughput* performance measure. The throughput is measured as the number of bytes sent divided by the time it took sending them. To be precise, the measured time is actually the time the service user is delivering data to the service provider. However, as the message sent is relatively large, the additional time the service provider needs to send data received from the service user after its return is minimal compared to the time it takes to send the whole message.

The *ttcp* application program [10] is used to measure the TCP memory-to-memory throughput. The *ttcp* program is modified to set the socket options which set the maximum TCP window size on a connection.

Each measured throughput point is the average of 25 runs. Each run transfers a 16-Mbyte message between the sender and the receiver. The message is composed of user data buffers. A user data

[2] *Million Instructions Per Second.*

buffer is an application SDU (service data unit). The word *segment* will be used about the transport PDU less the transport header. The throughput denomination is Mbit/s[3]. For measure points that do not experience packet loss, the coefficient of variation[4] mainly lies between 0.01 and 0.02. The maximum size of the TCP window is 65535 ($2^{16} - 1$) which constraints the maximum size of the socket receive buffer.[5]

2.3 Performance measurement environment

The performance measurements are based on the standard TCP/IP protocol stack in SunOS 4.1.x, derived from 4.2 BSD and 4.3 BSD. The two different end system architectures are the Sparc2-based Sun IPX and the Sparc10-based Axil 311/5.1. The MIPS rating is about 4.5 times higher for the Sparc10 compared to the Sparc2 (135.5 vs. 28.5). The SPECint92[6] is 65.2 vs. 21.8. The Sparc2s are equipped with 16 Mbytes main memory, and the Sparc10s with 32 Mbytes main memory. Apart from the CPU, these machines clearly differ in their bus and memory architectures. However, the I/O bus of both machine architectures is the SBus [11] to which the network adapter is attached. The Sun machines run SunOS 4.1.1, while the Axil machines run SunOS 4.1.3. The differences between the two SunOS versions are negligible in the performed performance measurements. Access to the TCP protocol is through the BSD-based socket API [12].

The ATM network interfaces are the first and second generation ForeRunner™ ATM SBus interfaces from FORE Systems Inc. running network driver version 2.2.6. They support both switched and permanent virtual circuits.

The first generation SBA-100 SBus ATM adapter [13] [14] is a simple slave-only interface based on programmed I/O (PIO). The ATM interface has a 16-kbyte receive FIFO (First In First Out) and a 2-kbyte transmit FIFO. The SBA-100 network adapter performs on-board computation of the cell based AAL3/4 CRC, but the segmentation and reassembly between frames and cells are done entirely in software by the network driver. The second generation SBA-200 SBus adapter [15] [16] includes an embedded Intel i960 RISC control processor, and hardware support for both AAL 3/4 and AAL5 CRC calculation. It is an SBus master device and uses direct memory access (DMA) for data transfer on both the send and receive path. ATM segmentation and reassembly are performed by the control processor on the adapter using the host memory for storage of frames.

For a more detailed description of the performance measurement environment, see [3].

3 Application performance factors

The application controls the size of two kinds of buffers which are significant to measured performance, namely, the user data buffers and the socket buffers. The size of the user data buffers affect the number of system calls which are required to send or receive an application message. The smaller this buffer size, the higher the number of system calls.

The application gets access to the communication services through the socket API. The API is a collection of system calls. The application data is copied to socket layer buffers before the underlying communication protocol TCP starts handling the data to be transmitted. The socket buffers limit the number of outstanding unacknowledged bytes. Both the copy operation and mechanisms of TCP affect the segmentation of the application byte stream; TCP is a byte-stream oriented protocol which does not preserve application level user data buffer boundaries between the two communicating peers.

3.1 The application user data size

The send user data buffer size decides the number of system calls to send a specific amount of data. The size of the receive user data buffer limits the number of bytes to be received within one system call. The required number of system calls to receive a specific amount of data may vary as the receive system calls may return with a non-full buffer. The size of the receive buffer may also affect the rate at which window updates may be returned.

For a given number of bytes to be transferred, the smaller the user data size of the send system call, the more system calls need to be performed. This limits the achievable throughput in two ways. One is due to the processing resources to do many system calls. The other is due to a lower average size of the protocol segments which are transmitted on the connection. Therefore, for small user data sizes, increasing the user data size will increase the throughput. The increase in throughput flattens when an increase in user data size does not significantly influence the average segment size on the connection.

For small window sizes there are characteristic throughput peaks dependent on user data size. For an 8-kbyte window the variation can be as high as 10 Mbit/s for different user data sizes, Figure 2. This is caused by a mismatch between user data size and window size which directly affects the segment flow. Figure 2 shows that the throughput pattern to some extent repeats for every 4096 bytes. Window and user data sizes that result in the least number of segments give the highest performance.

For the largest window sizes there are less characteristic peaks in the throughput graph. The shape of the graph is more like a sawtooth pattern with the throughput peaks at user data sizes which are an integer multiple of 1024 bytes. At such user data sizes the memory utilization is better because each operating system mbuf [12] will be completely filled. Consequently, sending a packet is slightly faster as it is composed of less data buffers. This gives a small increase to an otherwise saturating throughput level. For the larger window sizes, the number of outstanding unacknowledged bytes seldom reaches the window size. The segment flow is primarily maximum

[3] *1 Mbit/s = 106 bit/s, 1 Mbyte = 1024*1024 bytes, 1 kbyte = 1024 bytes.*

[4] *Standard deviation divided by mean.*

[5] *In SunOS 4.1.x the maximum allowed socket buffer size is 52428. In the following measurements this is changed to the TCP maximum window size. The applied socket buffer sizes are 4096, 8192, 16384, 24576, 32768, 40960, 49152, 52428, 57344, and 65535.*

[6] *The SPEC benchmark evaluates performance by measuring the elapsed time of each of 10 benchmarks.*

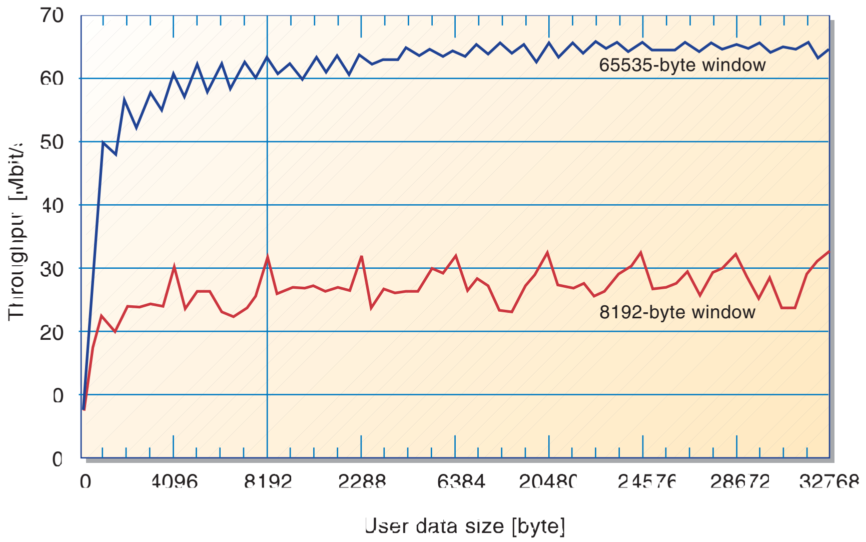


Figure 2 TCP throughput dependent on user data size [TCP/IP over AAL5, Sparc10 DMA (SBA-200/2.2.6) sub.]

sized segments of 9148 bytes with an acknowledgment returned for every other segment.

3.2 The memory address alignment of the user data size

The socket API transfers data between the user and system domains by explicitly copying the data. Thus, the two domains use different buffer management schemes which both retain complete control of use and reuse of their buffers. The buffers of the application may have arbitrary memory address alignment and be of arbitrary length. The network buffer memory management in the kernel uses chains of mbufs [12] [17].

The arbitrary alignment of the application buffer affects the efficiency of the copy operation and thereby the efficiency of the API. This is illustrated in Table 1 which for two different host architectures, Sparc2 and Sparc10, presents the average time of 1000 moves of 9148 (MSS) bytes from the user buffer to a chain of mbufs on send, and the other way around on receive. The start of the user buffer is “Offset” bytes above a 16-kbyte aligned memory address. The efficiency of the socket API is the highest when the application buffer starts on a double word boundary. Probably, the reason is the use of double loads and stores which also is a good match to the 8-byte wide write buffer between the (first level) cache and the next level of the memory hierarchy; main memory (Sparc2), and second level external cache (Sparc10)[7].

4 Protocol performance factors

The communication service offered the application obviously depends on the chosen protocol stack. The higher the layer and the more advanced the offered service, the less is the measured throughput. Protocol processing overhead can be divided into packet- and byte-related overhead. The *per-packet overhead* is related to the processing of the protocol header itself and is independent of the size of the packet, e.g. the demultiplexing of incoming frames to their higher-layer protocol. The *per-byte overhead* depends on the size of the packet and usually increases linearly with the number of data bytes to manipulate, e.g. calculation of a checksum for error protection and detection.

The TCP transport layer may segment or block the user data application SDUs. TCP sets a maximum segment size (MSS) allowed on the TCP connection. The TCP MSS is computed as the network maximum transmission unit (MTU) less the 40-byte standard TCP/IP header. In addition to the above factors, i.e. the user data size of the application interface and the MTU of the underlying network

Table 1 Socket API copy time depends on the buffer address alignment 9148-byte user data buffer, 16 kbyte + offset alignment

(a) Sparc2

Offset	0	1	2	3	4	8	12	16	20	24	28	32
Send copy [µs]	568	1098	1091	1089	767	539	756	540	745	533	766	541
Receive copy [µs]	517	1189	1154	1162	847	524	835	505	833	506	843	505

(b) Sparc10

Offset	0	1	2	3	4	8	12	16	20	24	28	32
Send copy [µs]	309	412	408	412	318	227	244	241	250	237	288	289
Receive copy [µs]	118	379	380	380	239	112	192	111	202	108	195	117

[7] *Somehow, the Sparc10 send is higher for a 0-byte offset than the other 8-byte aligned addresses. Correspondingly, ttcp measurements using a user buffer aligned at a 16k+8 byte memory address increased the maximum performance from 65 to 72 Mbit/s, that is about 11 %.*

technology, the operating system implementation environment and the host architecture affect the TCP segment flow [18] [19].

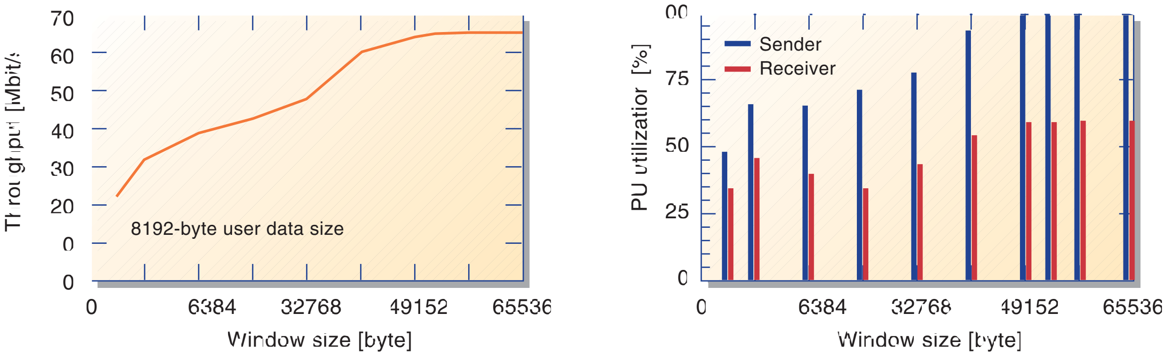


Figure 3 Throughput and CPU utilization [TCP/IP over AAL5, Sparc10 DMA(SBA-200/2.2.6) sub.]

4.1 TCP protocol options

Several extensions have been suggested to make TCP perform better over high-capacity paths and paths with a high bandwidth-delay product. Backward compatible extensions are implemented through the use of new TCP options, and additional connection state information:

- *TCP window scale option* [20] [21] Expanded window size to allow windows larger than 65535 bytes by introducing an implicit scale factor. The TCP window field in the header multiplied by this implicit scale factor gives the actual window size. The combination of a high transmission capacity and an unchanged signal propagation speed (the speed of light) is a primary issue in end-to-end communication performance in wide area networks. The higher the bandwidth-delay product[8], the higher the application's sensitivity to a long propagation delay. Within local area networks, the bandwidth-delay product is less of an issue as the delay factor is at a minimum.
- *TCP time stamp option* [21] [22] A segment time stamp which is returned by the receiver in the corresponding acknowledgment allows the sender to compute the round-trip delay as the difference between the current time and the time stamp. This gives a better round-trip time estimate that does not include retransmitted segments. The segment time stamp may also be used to avoid duplication of sequence numbers which may occur with a large window size and the wrap-around of the 32-bit TCP sequence number field. The basic idea is that if the arriving segment time stamp is less than the time stamp of the most recently accepted in-sequence segment, the segment can be discarded as an old duplicate.
- *TCP selective acknowledgment* [20] [23] The use of selective instead of cumulative acknowledgment avoids retransmitting a large data volume of which most actually did reach the receiver. The sender retransmits only the segments that have actually been lost.

Additional experimental extensions have been specified to fill the gap between the connection-oriented TCP and the datagram oriented UDP:

- A partial order service [24] to be used in e.g. multimedia applications and distributed databases. This service allows the applications to lower the demanded Quality of Service of the communication assuming that such a service is more efficient and less costly.
- An effective transaction oriented (request-response) service [25] to make use of the TCP connection-oriented service with reliable delivery, but with an accelerated connection set-up and close.

4.2 Turning various TCP protocol parameters

This section focuses on TCP mechanisms which affect the size of the segments on a connection including the window size, acknowledgment strategy, the use of Nagle's algorithm, as well as the maximum segment size. The TCP flow control and acknowledgment strategy also limit the number of outstanding unacknowledged bytes. Nagle's algorithm [7] makes the size of the TCP segments to be primarily MSS bytes. On the contrary, the action sequence of the SunOS 4.1.x operating system on acknowledgment reception may let segments of size less than MSS bytes be transmitted [3].

4.2.1 TCP window size

The maximum TCP window size, which is announced by the receiver, equals the socket receive buffer size. Independent of the window size, the socket send buffer limits the number of outstanding unacknowledged bytes because this buffer is used as a repository for such. The maximum *effective* window size and its achievable throughput depends on the size of both the send and receive socket buffers.

Generally, the larger the window size, the higher the measured throughput, Figure 3. This is as expected, as an increase in window size will utilize more of the processing capacity of the hosts, and more data may be in transit between the sender and receiver. When none of the communicating end systems are overloaded, the window size is the bottleneck.

4.2.2 Turning off Nagle's algorithm

The average segment size is reduced when Nagle's algorithm is turned off. The measured throughput with and without Nagle's algorithm is presented in Figure 4 (a). Figure 4 (b) presents the CPU utilization without Nagle's. The user data size is 8192 bytes:

[8] *The bandwidth-delay product is the network transmission speed times the round-trip delay.*

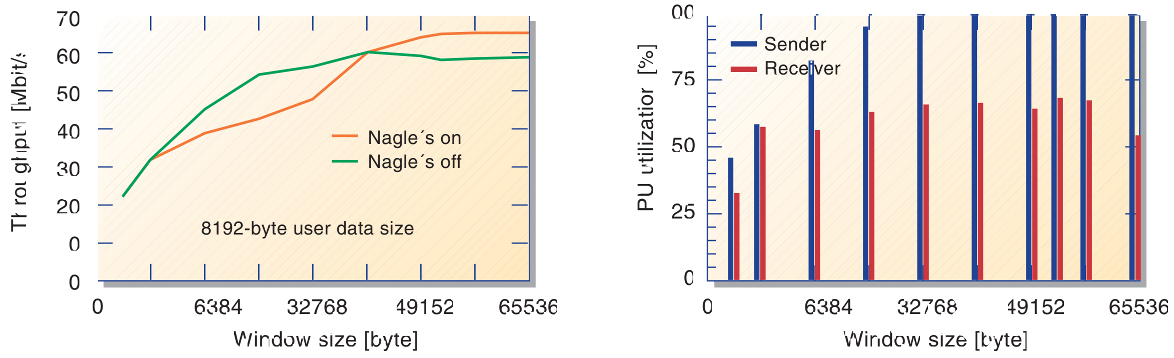


Figure 4 Throughput and CPU utilization, Nagle's algorithm turned off [TCP/IP over AAL5, Sparc10 DMA(SBA-200/2.2.6) sub.]

- For the smallest window sizes there is no change of performance, because turning off Nagle's algorithm does not affect the segment flow.
- For the medium sized windows, the window is now better utilized when smaller segments are permitted. This gives a throughput increase. The segment flow is primarily 4096-byte segments, and the processing load is better distributed on the sender and the receiver.
- For the larger window sizes the throughput is reduced by almost 10 % due to the increased overhead when allowing smaller packets on the connections.
- The relative CPU load difference is smaller compared to when Nagle's algorithm is in effect. Thus, using more packets to transmit a message of data increases total protocol processing overhead relatively more on the receiver than on the sender.

4.2.3 Explicit acknowledgments

The measurements presented in this section are run with the SunOS kernel compile option that turns off the delayed acknowledgment scheme, a scheme that relies on window updates being returned after data has been moved to user space. On the contrary, the receiver now returns an acknowledgment before the application in user space copies data to its buffers. TCP will acknowledge each and every incoming segment, so for large windows the sender transmits MSS bytes in-between received acknowledgments.

Figure 5 (a) and (b) present the throughput and CPU utilization, respectively. To ease the comparison, the throughput when using the delayed acknowledgment scheme is also presented:

- For the smallest window sizes there is no change of performance, because altering the acknowledgment scheme does not affect the segment flow.
- For medium sized windows, there is a clear increase in throughput. Thus, a faster returned acknowledgment relieves the sender from waiting on a window update before the next segment can be transmitted. This is especially evident for a 32-kbyte window size which has a throughput increase of about 35 %.
- The larger window sizes experience a small throughput degradation.

From the above it is evident that for certain window sizes it is advantageous to turn off the delayed acknowledgment strategy. However, as this is a kernel *compile* option in SunOS 4.1.x, all TCP connections are set to acknowledge all incoming segments – also on networks with a smaller MTU such as Ethernet.

For all except the smallest window size, the difference between the CPU load of the sender and receiver is smaller than when using the delayed acknowledgment scheme. The increased number of acknowledgments put more or less an equal additional load on the sender and receiver; the byte-independent work of an incoming acknowledgment corresponds to the work of generating the acknowledgment. A more significant reason for the relatively heavier loaded receiver, is the wasted work which is

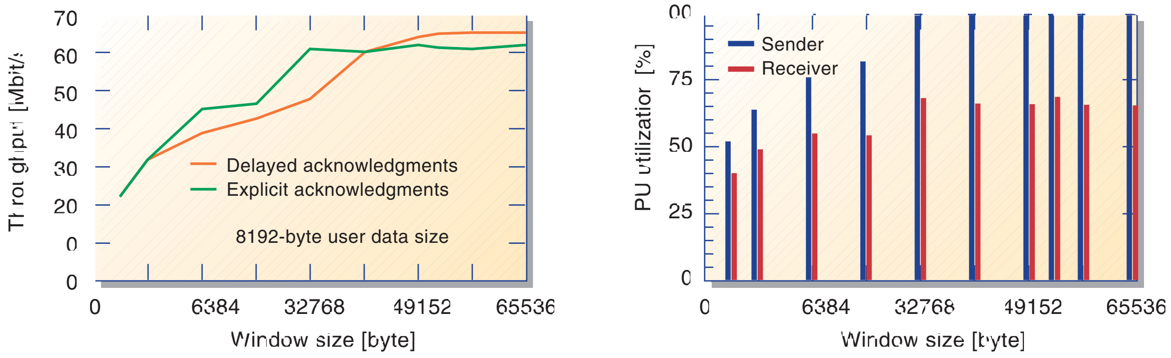


Figure 5 Throughput and CPU utilization, explicit acknowledgments [TCP/IP over AAL5, Sparc10 DMA(SBA-200/2.2.6) sub.]

required to check if a window update is to be returned after data has been moved to user space. This is seldom required, as window updates are announced by the explicit acknowledgments.

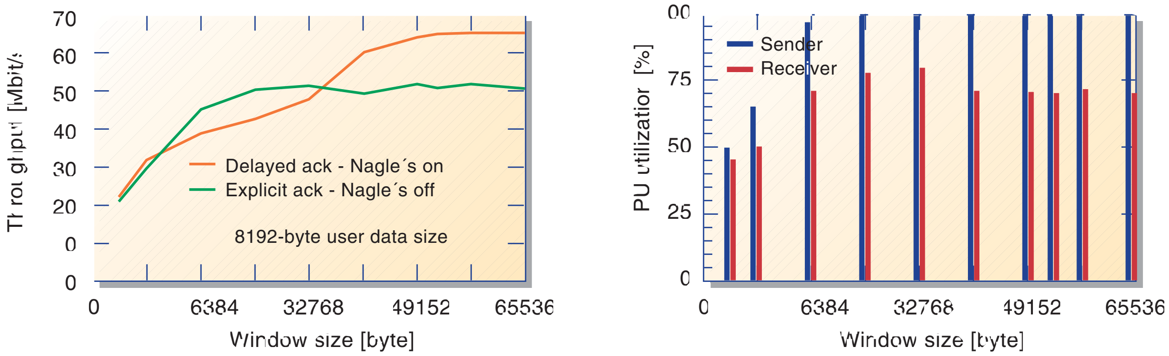


Figure 6 Throughput and CPU utilization. Nagle's algorithm turned off and explicit acknowledgments [TCP/IP over AAL5, Sparc10 DMA(SBA-200/2.2.6) sub.]

4.2.4 Turning off both Nagle's and the delayed acknowledgment

Figure 6 presents the measurements when both Nagle's algorithm and the TCP delayed acknowledgment strategy are turned off. The sender saturates at an even lower window size, and the maximum throughput is reduced by more than 15 %. This is as expected. Turning off Nagle's algorithm increases the number of segments which is necessary to transmit the specific amount of data. Thus, more segments are to be acknowledged as the delayed acknowledgment also is turned off. For Nagle's turned off, the receiver should use the delayed acknowledgment strategy for maximum performance.

The difference between the processing load at the sender and receiver is even smaller. This is as expected, because both Nagle's algorithm and the explicit acknowledgment scheme put a relatively heavier load on the receiver than on the sender.

4.2.5 Changing the TCP MSS size

The TCP MSS is computed from the network MTU by subtracting the 40-byte TCP/IP header. Thus, by changing the MTU size, the TCP MSS is also changed. The larger the MSS the larger the data chunk which is lost in case of cell loss, and the higher the probability of overrunning buffers, e.g. switch buffers, when the MSS chunk is transmitted as a burst of cells. Therefore, the MSS should not be indefinitely large.

Figure 7 presents the measured throughput dependent on MSS and window size. The end system configuration uses Nagle's algorithm and the delayed acknowledgment scheme. The MTU size is set so that the TCP MSS becomes an integer multiple of 1024 bytes. The user data size is 8192 bytes. The throughput achieved for corresponding measurements where the MTU size itself was an integer multiple of 1024 bytes, was up to 10 % lower. The maximum possible MTU size supported by the 2.2.6 driver is 16408 bytes [3] – too small to support a 16384-byte MSS. From the measurements it is evident that:

- The optimal size of the MSS depends on the window size.
- The larger the MSS, the larger the required window size to achieve a higher throughput than the corresponding throughput for a smaller MSS.
- For medium sized windows the smaller MSSs give the highest throughput. This is due to a better load distribution between the sender and receiver, and the fact that the average time to acknowledge the data bytes is shorter. A smaller MSS causes a window update, which slides the window two MSS bytes, to be returned more often. This has a positive effect on the measured throughput.
- An MSS of 8192 bytes performs as least as good as the default 9148-byte MSS. Thus, when the window size is the bottleneck, it may be advantageous to streamline the MSS with the window size.

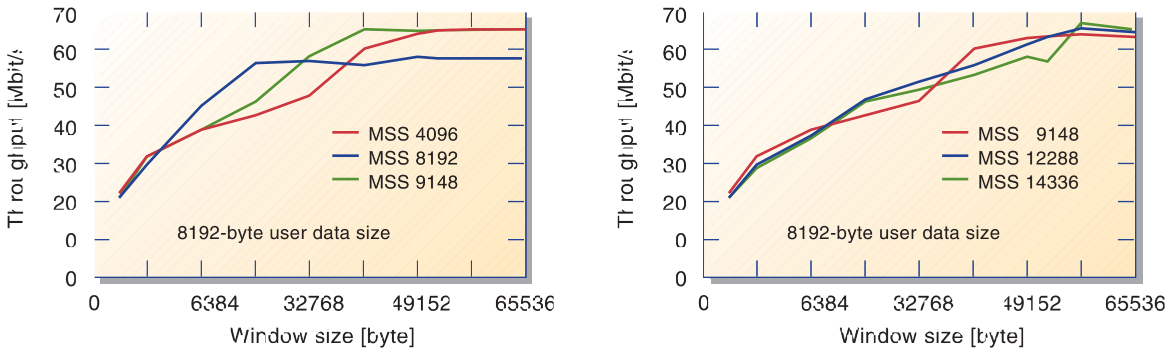


Figure 7 Throughput dependent on MSS, delayed acknowledgments [TCP/IP over AAL5, Sparc10 DMA(SBA-200/2.2.6) sub.]

As presented above running TCP protocols on machines attached to new network technologies requires cautious configuration of variables and parameters to be successful. In addition, a deficient deadlock behavior of TCP has been shown to be considerably more striking to achievable performance when TCP is run over high-speed networks with large maximum transmission units [3] [26] [27].

5 Operating system performance factors

The operating system constitutes the software environment in which the protocol stack is implemented. Operating system services used by protocol implementations are among others: system calls, protection domains, control structures, interrupts, context switching, memory sharing, copy operations; and timer, buffer, and memory management [9] [28] [29] [30] [31] [32] [33]:

- *System calls* are used to get access to operating system services which execute in a protected system domain. In operating systems with a monolithic kernel such as Unix, the API constitutes of a collection of relevant networking system calls to set-up and close connections, send and receive data, as well as set connection parameters. The system calls copy data between the user application and the operating system kernel in which the protocols are executed.
- An *interrupt* is used to signal an event which requires immediate attention. During the processing of the actual interrupt, all lower priority interrupts are blocked. The highest priority interrupt is the hardware clock interrupt which drives the system. A *network* interrupt signals the arrival of a packet. The network and transport layer protocol processing of incoming packets may be performed at a lower *software* interrupt level.
- A *process* is a task with its own address space. A *context switch* assigns another process than the running the right to execute, e.g. when a process executes for the whole duration of its time slice, or when the system identifies a higher-priority process to run. A context switch is a costly operation [34]. An implementation architecture where each protocol layer consists of one or more processes, would require many more context switches and be less efficient than an implementation architecture of which a process follows a packet through several layers of the protocol stack [31] [35]. While the service of interrupts and involuntary context switches appears asynchronously, a voluntary context switch appears synchronously with respect to the current process. A sending process performs a voluntary context switch if it has to await an acknowledgment before more data can be transmitted. Correspondingly, a receiving process performs a voluntary context switch if it has to await incoming data. The process is scheduled to run again when data has been received.
- *Timer management* is used by the protocol for instance to set retransmission timers or to periodically check if an acknowledgment should be returned.
- *Buffer memory management* deals with the system representation of SDUs and PDUs, while *general memory management* relates to the memory architecture of the end system.

Similarly to protocol processing overhead, operating system overhead is classified as either byte or packet oriented. The copying of data in a system call is clearly byte dependent. Packet-oriented overhead such as timer handling, context switching, and interrupt processing are typical operations that are performed independently of the packet size.

5.1 Host internal data movement

Several studies have shown that it is not the protocol processing, but the data manipulation tasks that are time consuming [9] [29] [36]. The introduction of RISC (Reduced Instruction Set Computer) machines enlarged the gap between the processor capacity and memory and bus bandwidths. A gap which continuously gets larger [37]. The *in*creasing performance gap between processor and memory bandwidth, and the *de*creasing gap of bus and network transmission speed, have made end system internal data movement an important factor in achievable performance.

An important factor of the end system bottleneck is the internal data movement among others when crossing the protection boundary between the user and the operating system kernel. An application programming interface which moves data by reference instead of by explicit copying has been shown to significantly increase throughput performance and at the same time reduce the processor load [3]. To further shorten the delay of traversing the protocol stack requires a tighter integration of the host and its network interface.

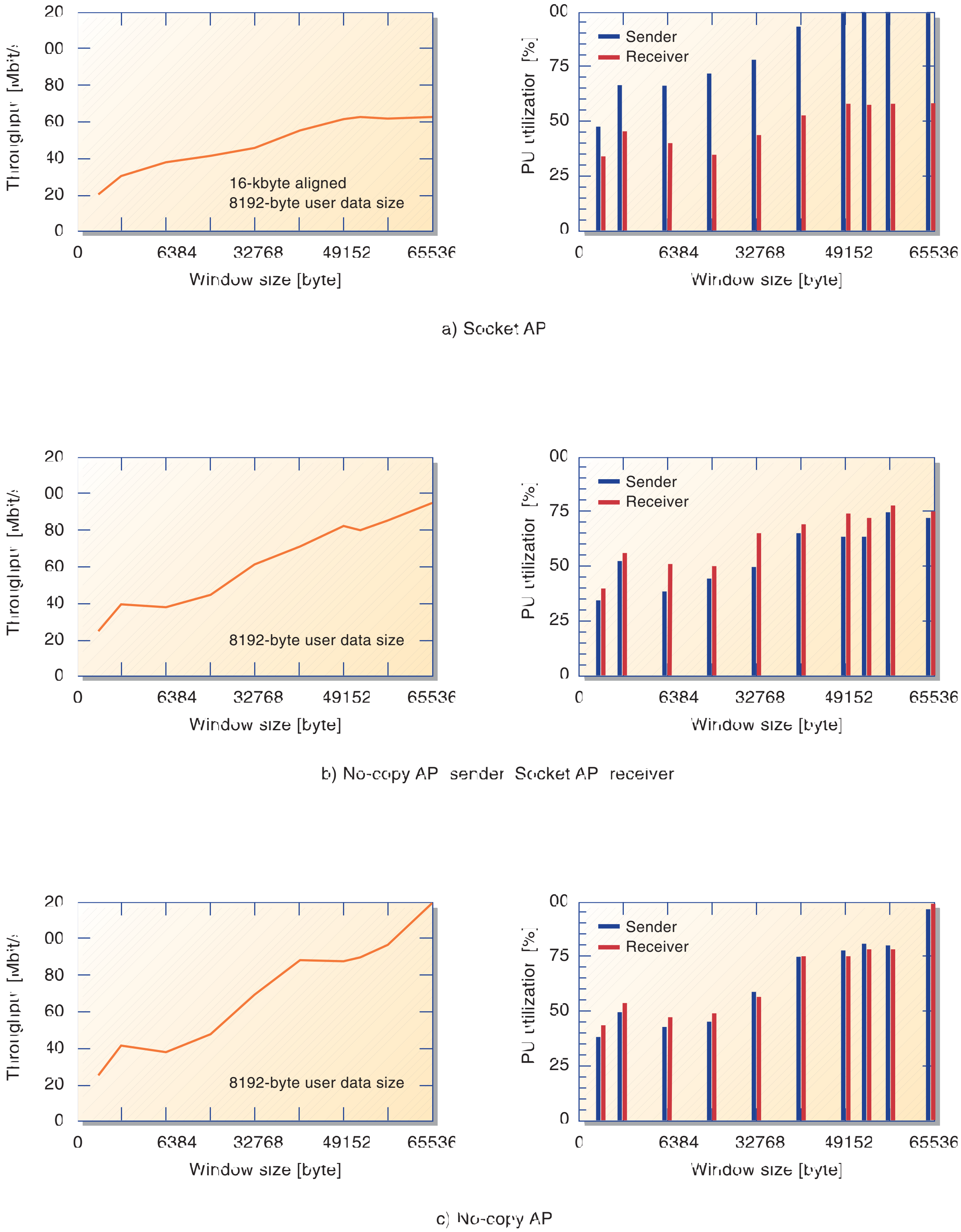
A network application has a contiguous or non-contiguous area in virtual memory of which to send or receive data. In BSD-based systems, e.g. SunOS 4.1.x, the *socket* layer provides an API of system calls towards the communication protocols.

The explicit data-copying *socket* API puts no restrictions on application buffer management. The explicit copy operation is performed due to consistency, protection and security issues. The copy operation translates the data to the network memory management structure of the kernel. When the transport protocol provides a reliable service over an unreliable network, a copy operation is required within the kernel for retransmission purposes. The end-to-end transport protocol may also compute a checksum on the transport PDU before calling the underlying network protocol. After the packet has been processed by the network protocol, it is queued at the network interface send queue. Dependent on the network adapter architecture, an additional copy operation may be necessary to collect fragments of a packet in contiguous memory (simple DMA). After the PDU has been transmitted out on the network wire by PIO or by DMA, it must be removed from the relevant interface send queue.

5.2 A no-copy API

When using the socket API, Figure 8 (a), the Sparc10 sender is clearly the bottleneck for the maximum 65-Mbit/s throughput. The introduction of a no-copy API [3] [38] at the sender, Figure 8 (b), results in a significant performance gain for the larger window sizes. The maximum achieved throughput is 94-Mbit/s.

Despite the window tending to become a bottleneck, the performance further improves when using the no-copy API both at the sender and receiver, Figure 8



a) Socket AP

b) No-copy AP sender Socket AP receiver

c) No-copy AP

Figure 8 Throughput and CPU utilization depends on API [TCP/IP over AAL5, Sparc10 DMA(SBA-200/2.2.6) sub.]

(c). The reason is a shorter round-trip time, making acknowledgments to be returned faster. The measured maximum throughput is 115 Mbit/s. The CPU utilization of the Sparc10 sender and receiver is now more or less the same. It approaches 100 % only for the largest window.

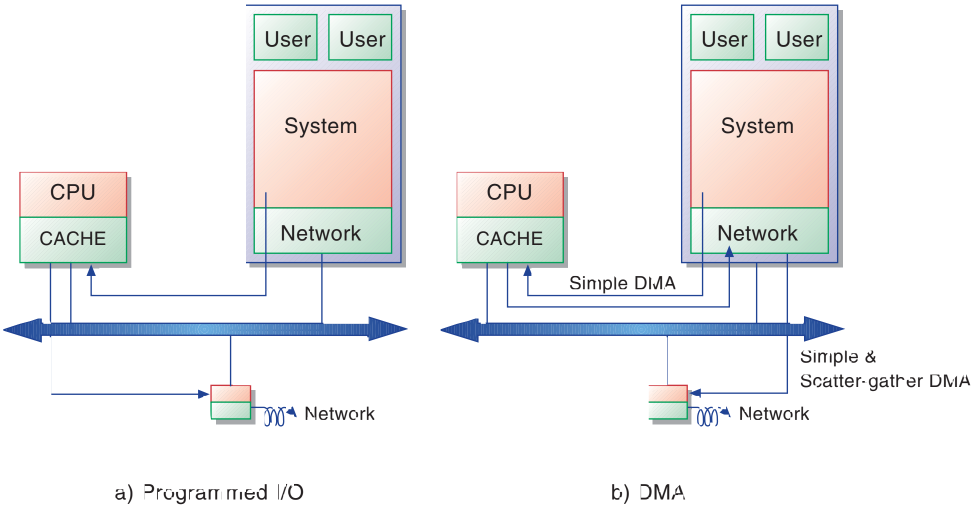


Figure 9 Data movement to the network adapter

5.3 Integrated layer processing

The idea of integrated layer processing (ILP) [4] is to minimize the time consuming data manipulation steps including moving data to and from the network, buffering data for retransmission, detecting data errors, encrypting and decrypting data, moving data to/from application address space, and formatting data for presentation. A key to achieve ILP is application level framing (ALF) which provides a single common data unit over which manipulations are defined.

Integrating data manipulation operations from different layers in one or two processing loops, implies leaving the structured and modular layered implementation of current communication protocols; sharing fields and structures across different layers is permitted. The result is an implementation structure which may be difficult to maintain. Further problems applying to ILP is that traditionally imposed precedence or ordering constraints may limit the opportunities for integrated layer optimization; checksumming the user data cannot be performed before the data units of the segment to be checksummed are in order, and different data manipulations require different sized data; some may even change the quantity of data.

The TCP/IP implementation of Solaris 2 combines the two-pass process that involves calculating the TCP checksum and copying data between the kernel and the user application into one single pass. By eliminating one entire pass over the user data, this *integrated copy and checksum* technique [39] significantly improves the raw TCP/IP data throughput. This is evident from Figure 10.

6 Network performance factors

The achievable throughput strongly depends on the architecture of the ATM network interface. An upgrade of the network interface to relieve the host of cell-related processing and of moving the data between main memory and the network adapter, is shown to be more valuable to a higher performance than an upgrade of the host processing capacity, Figure 10. The performance effect of a host upgrade is clearly lower than the difference in host computational performance rating may seem to promise.

6.1 Host integration of ATM network interfaces

The nature of ATM poses several challenging problems compared to the design of network interfaces for Ethernet, fast Ethernet, or FDDI. In addition to the connection oriented approach, highly ATM-specific issues are the cell oriented level of multiplexing, traffic management, and the potential for negotiated quality of service guarantees. To provide an interface towards the network that takes advantage of the qualities of ATM, the network adapter must concurrently pay attention to all opened ATM connections. General considerations that affect the ATM network interface integration into the host environment are:

- *The partitioning of functions*
  When the higher-layer protocol processing is done by the host, the network interface mainly supports a link-layer service interface. If the network adapter supports complete on-board protocol processing, the programming interface may either be the transport layer or at the application level. In-between these interfaces, there are network adapters that support higher-layer data manipulation operations such as calculating a checksum.
- *The method of data movement*
  The abstraction of a network interface is a device which can transfer data between the network and the host's memory. Usually, the transfer itself is through the use of PIO or DMA. The most efficient data movement technique depends on the goal of the network interface; low latency or high-speed. To optimize the use of the system internal bus, burst transfers are profitable.
- *Network adapter flexibility*
  The more functionality that is put in software-programmable elements, the higher the flexibility of the interface. However, the desire for speed limits the extent to which processing on the adapter can be accomplished in software.
- *Exchange of status and control*
  The status and control information to be exchanged between the host and the network interface depends on all of the above considerations.

Any evaluation of a network interface design and its integration into the host is subject to the specifics of the host, its I/O bus, the network adapter technology and software, the network technology, as well as the target applications. When dealing with commercial development of network interfaces, additional important factors are related issues such as complexity, cost and time to market.

Outboard processing of higher-layer protocols makes the network adapter a front-end, possibly with its own operating system. This is more complex and expensive than a network interface providing a link-layer interface. The front-end relieves the host from higher-level protocol processing, but care must be taken to avoid the host-adapter interface from becoming a new bottleneck [40]. Protocol processing should not be removed from the host unless the non-protocol processing overhead is reduced as well [41]. Otherwise, the front-end will face the same efficiency problems as protocol processing on the host. The front-end processing capacity is often less than the host processing capacity, and any upgrade of the host is not taken advantage of. ATM network interfaces therefore provide an ATM adaptation link-layer interface.

6.2 Data movement between host memory and network adapter

There are two principle end system data paths on traversing a protocol stack. The host internal data movement between main memory and CPU; within or between addressing domains as discussed above, and data movement between main memory and network adapter to be discussed next.

There are two principle ways of moving data between host memory and network adapter, namely programmed Input/Output (PIO) in Figure 9 (a) and direct memory access (DMA) in Figure 9 (b). The two principle ways of moving data between the host memory and the network adapter are distinguished on the CPU's involvement in the actual byte transfer.

Data movement between host memory and network adapter, whether it is by PIO or DMA, inevitably results in data being transferred across the I/O bus; because data is moved between two physically separated units. The efficiency of the bus transfer mechanism depends on both the host architecture and the application. If the CPU itself can do burst transfers across the bus, this may for certain instances be in favor of PIO. How fast the application can access the requested data in each case is also of importance. Using PIO data is in cache after it is read from the adapter. To take advantage of this, the application should be scheduled before its data is replaced in the cache. Using DMA, data may be buffered in main memory allowing better dynamic sharing of memory resources between different processes and domains.

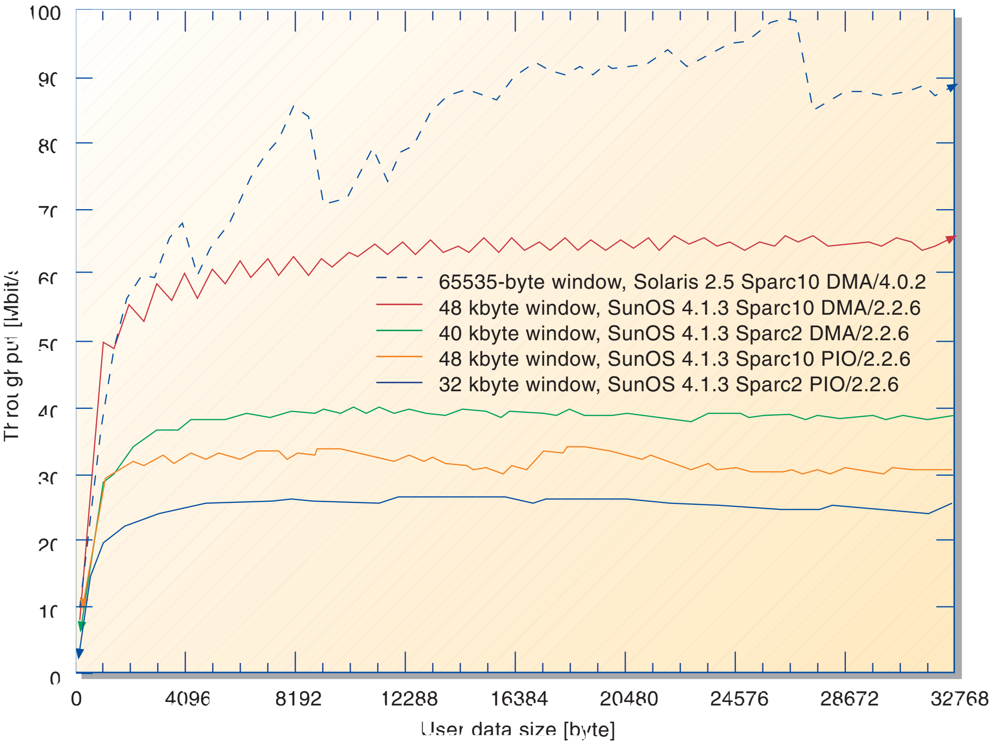


Figure 10 Maximum throughput dependent on end system configuration

Figure 10 presents throughput measurements for both DMA and PIO network interfaces for the saturating window size for the different configurations. The measurements show that:

- The network adapter should relieve the host of any cell-oriented processing, as well as of the data movement between the host and the network adapter.
- The measured throughput clearly depends on both the host architecture and the network interface. An upgrade of the network adapter may be more valuable to a higher throughput than an upgrade of the host processing capacity.
- Even if the size of the window is the bottleneck, the achieved throughput depends on the host and network interface. The smaller the byte-dependent overhead in the protocol processing and network driver, the larger the variations for a given window size.
- Which end system, the sender or receiver, is the bottleneck depends on the network interface. When using the PIO-based SBA-100 network adapters, the receiving end system is clearly heavier loaded than the sending end system. When using the DMA-based SBA-200 network adapters, the sender is heavier loaded than the receiver. The relative difference between load on the sender and receiver is affected by the cost of integrating the network adapter into the host architecture.

7 Summary and conclusions

This paper has presented issues related to end system networking performance over high-speed local area networks. Exploration and evaluation of the achievable performance are conducted through extensive performance measurements for different configurations of end system hardware and software. Modules in focus are ATM network interfaces, the Internet transport protocol TCP and its implementation within the SunOS Unix operating system, the application programming interface, as well as host architectures. The performance is evaluated in terms of throughput and processor utilization.

The performance bottleneck is the end system which challenge is the implementation structure and architecture of the communication protocols. This tends to be even more important than the protocols themselves. Consequently, achievable networking performance is not only a matter of fast transmission networks, but more a matter of application data unit size, the application programming interface, protocol mechanisms, operating system buffer management, host processing and data movement capacity, and the network interface which gives the host access to the high-speed network.

References

1 Saltzer, J H, Reed, D P, Clark, D D. End-to-end arguments in system design. In: *Proceedings of Second International Conference on Distributed Systems,* April 1981, 509–512.

2 Feldmeier, D C. A framework of architectural concepts for high-speed communication systems. *IEEE Journal on Selected Areas in Communications,* 11, (4), 480–488, 1993.

3 Moldeklev, K. *Performance analyses and issues of end systems attached to high-speed networks.* Ph.D. Dissertation 1996:42, Department of computer Systems and Telematics 1996:4, April 1996. (ISBN82-7119-928-5.)

4 Clark, D D, Tennenhouse, D L. Architectural considerations for a new generation of protocols. In: *Proceedings of Sigcomm'90, Computer Communication Review,* 20, (4), 200–208, 1990.

5 Haas, Z. A protocol structure for high-speed communication over broadband ISDN. *IEEE Network,* 5, (1), 64–70, 1991.

6 Heidelberger, P, and Lavenberg, S S. Computer performance evaluation methodology. *IEEE Transactions on Computers,* C-33, (12), 1195–1220, 1984.

7 Nagle, J. Congestion control in TCP/IP internetworks. *Internet RFC 896,* January 1984.

8 Sjödin, P et al. Towards protocol benchmarks. In: H. Rudin (ed.) *Protocols for High-Speed Networks II.* North-Holland, pp. 23–33, 1989. (ISBN 0-444-88536-6.)

9 Clark, D D et al. An analysis of TCP processing overhead. *IEEE Communications Magazine,* 27, (6), 23–29, 1989.

10 SGI, available from ftp.sgi.com ~ftp/sgi/src/ttcp

11 Lyle, J D. *SBus : Information, applications and experience.* New York, Springer-Verlag, 1992. (ISBN 0-387-97862-3.)

12 Leffler, S J et al. *The Design and Implementation of the 4.3BSD UNIX Operating System.* Reading, Mass., Addison-Wesley, 1989. (ISBN 0-201-06196-1.)

13 Cooper, E et al. Host interface designs for ATM LANs. In: *Proceedings of 16th IEEC Conference on Local Computer Networks,* Minneapolis, Minnesota, 14–17 October 1991, 247–258. ISBN 0-8186-2370-5.

14 FORE Systems. *SBA-100 SBus ATM computer interface : user's manual,* 1992.

15 FORE Systems. *200-Series ATM adapter : design and architecture,* January 1994.

16 FORE Systems. *ForeRunnerTM SBA-100/-200 ATM SBus Adapter : user's manual,* April 1994.

17 Sun Microsystems. *SunOS 4.1.1 Network programming guide,* March 1990.

18 Moldeklev, K. Klovning, E, Kure, Ø. TCP/IP behavior in a high-speed local ATM network environment. In: *Proceedings of the 19th IEEE Conference on Local Computer Networks,* Minneapolis, Minnesota, 2–5 October 1994, 176-185. (ISBN 0-8186-6680-3.)

19 Moldeklev, K, Klovning, E, Kure, Ø. The effect of end system hardware and software on TCP/IP throughput performance over a local ATM network. *Telektronikk,* 91, (2/3), 155–167, 1995.

20 Jacobson, V, Braden, B, TCP extensions for long-delay paths. *Internet RFC 1072,* October 1988.

21 Jacobson, V, Braden, B, Borman, D. TCP extensions for high-performance. *Internet RFC 1323,* May 1992.

22 Jacobson, V, Braden, B. TCP extensions for high-speed paths. *Internet RFC 1185,* October 1990.

23 Mathis, M B et al. TCP selective acknowledgment option. *Internet draft-ietf-tcplw-sack-00.txt,* January 1996.

24 Connolly, T, Amer, P, Conrad, P. An Extension to TCP : partial order service. *Internet RFC 1693,* November 1994.

25 Braden, B. T/TCP : TCP extensions for transactions, functional specification. *Internet RFC 1644,* July 1994.

26 Moldeklev, K, Gunningberg, P. Deadlock situations in TCP over ATM. In: G. Neufeld and M. Ito (eds.), *Protocol for high speed networks IV,* pp. 243–259. London, Chapman & Hall, 1994. (ISBN 0-412-71180-X.)

27 Moldeklev, K, Gunningberg, P. How a large ATM MTU causes deadlocks in TCP data transfers. *IEEE/ACM Transactions on Networking,* 3, (4), 409–422, 1995.

28 Braden, B, Borman, D, Partridge, C. Computing the Internet checksum. *Computer Communication Review,* 19, 86–94, 1989.

29 Kay, J, Pasquale, J. The importance of non-data touching processing overheads in TCP/IP. In: *Proceedings of Sigcomm'93, Computer Communication Review,* 23, (4), 259–268, 1993.

30 McKenny, P E, Dove, K F. Efficient demultiplexing of incoming TCP packets. In: *Proceedings of Sigcomm'92, Computer Communication Review,* 22, (4), 269–279, 1992.

31 Svobodova, L. Implementing OSI systems. *IEEE Journal of Selected Areas in Communication,* 7, (7), 1115–1130, 1989.

32 Zhang, L. Why TCP timers don't work well. In: *Proceedings of Sigcomm'86, Computer Communication Review,* 16, (3), 397–495, 1986.

33 Woodside, C M, Montealegre, J R. The effect of buffering strategies on protocol execution performance. *IEEE Transactions on Communications,* 37, (6), 545–553, 1989.

34 Mogul, J C, Borg, A. The effect of context switches on cache performance. In: *Proceedings Fourth International Conference on Architectural Support for Programming Languages and Operating Systems,* April 1991, 75–84.

35 Watson, R W, Mamrak, S A. Gaining efficiency in transport services by appropriate design and implementation choices. *ACM Transactions on Computer Systems,* 5, (2), 97–120, 1987.

36 Cabrera, L-P et al. User-process communication performance in networks of computers. *IEEE Transactions on Communications,* 14, (1), 38–53, 1988.

37 Patterson , D A, Hennessy, J L. *Computer Architecture : a quantitative approach.* Palo Alto, CA, Morgan Kaufmann, 1990. (ISBN 1-55860-188-0.)

38 Ahlgren, B, Björkman, M, Moldeklev, K. The performance of a no-copy API for communication. *IEEE Workshop on the architecture and implementation of high performance communication subsystems.* Mystic, Connecticut, 23–25 August 1995.

39 Sun Microsystems. *Solaris White Paper TCP/IP*. February 1996.

40 Cooper, E C et al. Protocol implementation on the Nectar communication processor. In: *Proceedings of Sigcomm'90, Computer Communication Review,* 20, (4), 135–144, 1990.

41 Steenkiste, P A. Analyzing communication latency using the Nectar communication processor. *Proceedings of Sigcomm'92, Computer Communication Review,* 22, (4), 199–209, 1992.

Kjersti Moldeklev is Research Scientist at Telenor R&D, Kjeller.

email address:
kjersti.moldeklev@fou.telenor.no

High speed networking in Internet and corporate Intranets

ESPEN KLOVNING AND HAAKON BRYHNI

1 Introduction

Until recently, Internet was an information medium used primarily within academic circles. The available services were primitive (e.g. telnet, email, ftp) but effective. With the introduction of WWW technology this situation has changed. The interest in Internet outside academic circles has exploded, commercial services are becoming available, e.g. banking, shopping, entertainment. Organizations, companies and cooperating parties are building Intranets to take advantage of the technology and its possibilities.

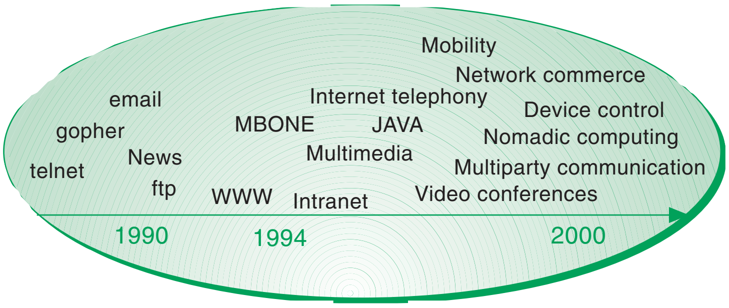


Figure 1 Internet service evolution

At the moment, both the number of users and the aggregated Internet traffic increases exponentially, and new services are made available continuously. Figure 1 presents an overview of the expected service development before the turn of the century. Over the next 5 years, available services on the Internet will include everything from network based entertainment to multimedia and nomadic computing.

To face up to the obvious challenges introduced by the Internet explosion, an upgrade of the Internet technology (e.g. IPv6 [1], RSVP [2], security [3], mobility [4]) and local area and wide area network infrastructure will be absolutely necessary. In this paper we will focus on the network infrastructure and two of the high-speed network technologies which will be used. The technologies we will study are Gigabit Ethernet and Asynchronous Transfer Mode (ATM).

The Gigabit Ethernet standard [5] is about to be completed within the IEEE 802.3z Working Group. It is backward compatible with the Ethernet standard which is the dominating networking technology in corporate networks today. The estimated Ethernet user group consists of more than 120 million users. In the last couple of years, additional technologies like Ethernet switching and 100 Mbit/s Fast Ethernet technology have been introduced to solve some of the limitations of the 10 Mbit/s shared medium Ethernet technology. Use of Fast Ethernet to the desktop requires an improvement in the transmission and switching capability in the local area backbone network. Gigabit Ethernet is targeted for high capacity backbones beyond the Fast Ethernet domain. However, the alliance sees Gigabit Ethernet as a future desktop technology when the end-systems are capable of utilizing the raw bandwidth.

ATM is a high-speed technology which oddly enough has been embraced by both tele- and data-communication groups. It offers a scalable network solution capable of multicasting with link bandwidths ranging from 25 Mbit/s to 622 Mbit/s to the desktop. Commercial products have been available since 1992 but ATM has still failed to take over as the preferred desktop network technology. The market growth is mainly in the local and wide area backbone network domain. The modest market growth is caused by several factors. First, the ATM port price is higher. Second, ATM lacks compatibility with the installed base of legacy LANs. Third, ATM is a complicated technology which still does not have all the pieces in place.

In this overview article we will discuss these technologies in an Internet and Intranet setting, i.e. local area networks, network servers (i.e. file servers, WWW servers, DB servers) and wide area networks. The article consists of four main sections. Section 2 describes the Gigabit Ethernet technology in detail, including the physical transmission technology and novel packet bursting schemes to enhance the network diameter. Section 3 discusses the use of ATM in Internet in general and the different solutions in detail. A detailed description of the core ATM technology itself is outside the scope of this article and interested readers are referred to the ATM literature, e.g. [6]. The last section, Section 4, presents some concluding remarks and predictions regarding the use of these important technologies.

2 Gigabit Ethernet

The Gigabit extension of the existing Ethernet and Fast Ethernet standards is currently being standardized by the IEEE 802.3z Study Group. The first steps were taken in November 1995 when the IEEE 802.3 Working Group commissioned its High Speed Study Group (HSSG) to come up with a proposal for a Gigabit Ethernet extension. The standardization process of Gigabit Ethernet is on schedule and the final approval is expected later this year according to the preliminary time-table.

Shortly after HSSG got its mandate, a vendor alliance was established. Its main objective is to accelerate the development and implementation of the IEEE 802.3z standard. As of January 1997, 95 hardware and software vendors have joined in on the effort to promote a Gigabit Ethernet standard and its networking products. All the major players in the Ethernet and Fast Ethernet marketplace have already joined as well as other companies in the high speed networking niche. IEEE expects pre-standard products to appear early 1997. This should enable the marketplace to evaluate the technology before the final standard and fully compliant products appear.

2.1 Technology

As mentioned earlier, Gigabit Ethernet is an extension of the popular 10 Mbit/s and 100 Mbit/s Ethernet standards. The most important requirement of this new technology will be to ensure full compatibility with the huge base of Ethernet installations world-wide. This means

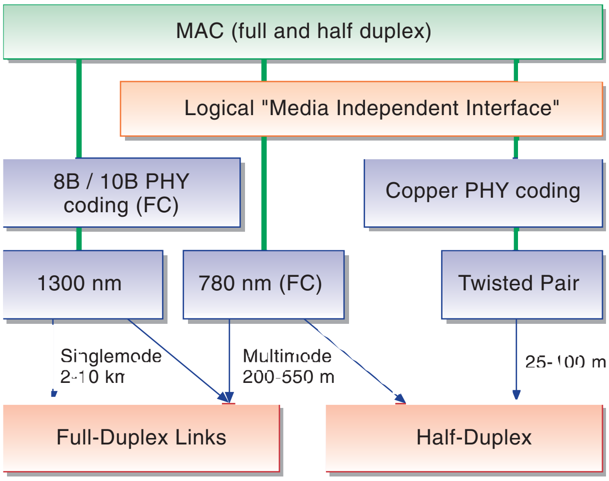


Figure 2 Gigabit Ethernet – Functional elements

supporting the CSMA/CD access method and the 1500 byte frame format used by Ethernet. Thus, Gigabit Ethernet will work seamlessly with the existing Internet technology and should be a plug-and-play Internet technology.

In addition to the half/duplex mode which supports shared connections using repeaters and the CSMA/CD access mode, Gigabit Ethernet will also support a full duplex mode. This mode will support switch interconnection and interconnection of switches and high-end end-systems.

Physical transmission technology

The physical transmission technology used by Gigabit Ethernet will use available and proven technologies to minimize the time-to-market. The functional elements of the Gigabit Ethernet standard is depicted in Figure 2.

The first version of Gigabit Ethernet will use an improved version of the physical transmission technology for Fibre Channel [16]. Fibre Channel is an ANSI standard for attaching high-performing I/O devices to computing devices, but it is also used for clustering computing devices, i.e. processors and workstations.

Currently, the physical transmission technology used by Fibre Channel is specified for a set of different transmission rates including 1062.5 Mbaud. This physical transmission technology will be enhanced to run at 1250 Mbaud to provide the required 1000 Mbit/s data rate. The optical components will use different wavelengths, i.e. 780 nm and 1300 nm, as well as both multi-mode and single-mode fibre. The optical emitters can be long-wave LEDs, short- or long-wave lasers. The transmission code will be the 8B/10B encoding/decoding scheme [17] which provides acceptable transition density and adequate DC balance for the optical receivers.

As illustrated in Figure 2, Gigabit Ethernet will not only use optical components but is also targeted for coax cables and eventually unshielded twisted pair category 5 (UTP5) cables. These physical layers require a new logical interface between the MAC and PHY layers in order to decouple the MAC layer from the specifics of the physical layers. This logical media independent layer interface is also illustrated in Figure 2. This will enable the PHY layer to use other coding techniques than the 8B/10B coding since the standard transmission coding for Fibre Channel transmission is not necessarily cost effective for UTP5. It should be noted that the use of UTP is not yet possible. However, the standardization group within IEEE 802.3z rely on expected advances in digital signal processing and in silicon technology to make it feasible.

Network diameter

Figure 2 illustrates that different cable technologies will support different link distances and network diameters. The limitations will be 25–100 m for UTP5 cabling and 200–550 m for MM fibre. With Fibre Channel, an SM fibre can support links up to 10 km. However, the standardization group for Gigabit Ethernet requires only support for link lengths up to 2 km.

One of the major issues within the IEEE 802.3z standardization is the network diameter. The 512 bit slot time of the original Ethernet specification limits the network diameter to 2.5 km including no more than 4 repeaters. With Fast Ethernet, the 512 bit slot time reduces the theoretical network diameter to one tenth, i.e. 250 m. It is obvious that this slot time cannot be used for a Gigabit Ethernet with a network diameter of approx. 200 m. A possible solution is to extend the minimal frame size from 512 bit to 512 byte, see Figure 3. The problem with this approach is that, 1) the maximum packet rate for small packets on a full duplex link will be lowered, 2) bandwidth would be wasted if the enlarged frames with a lot of padding is sent out on 10 or 100 Mbit/s Ethernets, and 3) padding and stripping is impossible in the 802.3 MAC when protocol stacks using the Ethernet frame format are used. The solution which has be chosen is to extend the carrier event on Gigabit Ethernet links using the CSMA/CD access method. The frame format of the proposed principle is illustrated in Figure 3 which also indicates the extended duration of the Carrier Event.

The extension symbols are non-data symbols which can be recognized by the PHY and MAC layers. The extension will be byte aligned and its length will be an integer number of bytes. Since the Frame Check Sequence (FCS) will not cover the extension but the Carrier Event includes it, this approach will work seamlessly with the existing Ethernet standards. The obvious drawback of this solution is that small packets might need as much as 448 byte (i.e. 512 – 64) more padding on a Gigabit Ethernet than on Fast Ethernet.

To increase the utilization, Gigabit Ethernet includes a novel CSMA/CD feature denoted packet bursting. This scheme allows any Gigabit Ethernet device to keep control of the network while a burst

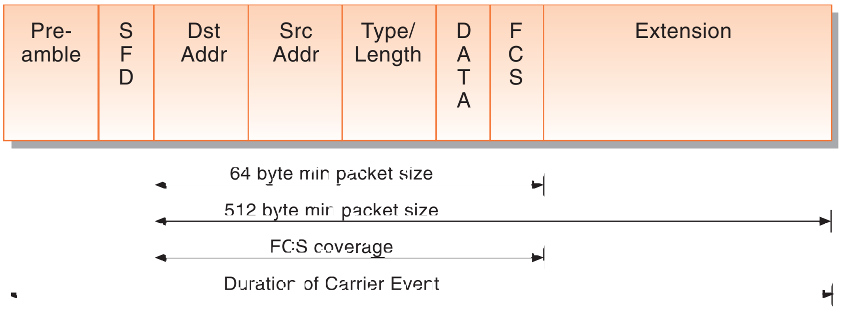


Figure 3 Carrier extension technique

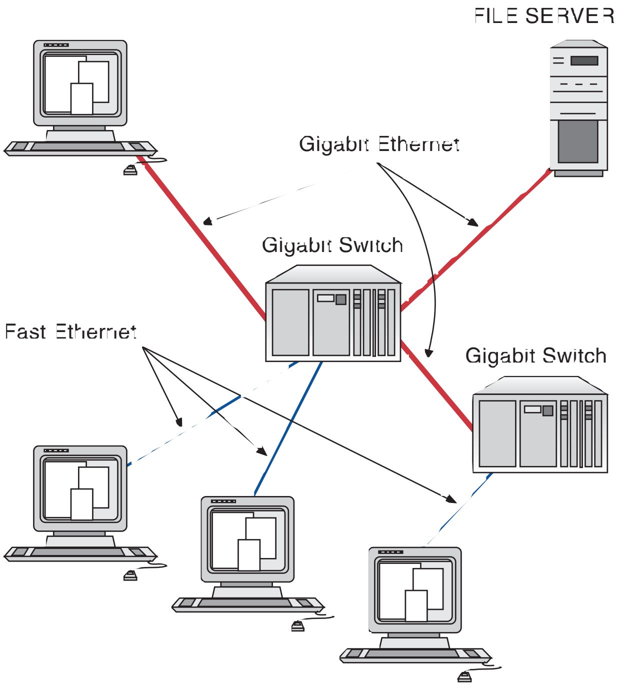


Figure 4 Gigabit Ethernet deployment

of small packets are transmitted over the network. The Carrier Event will not be disabled until all the packets have been successfully transmitted. If a collision occurs, a jamming signal will inform all recipients of all the previously sent packets in the burst about the collision. Packet bursting will increase the utilization in situations where a number of packets can be sent over the network during a single Carrier Event. If the host is not fast enough to use packet bursting, Gigabit Ethernet will offer backward compatibility at the expense of an extended Carrier Event which will lower the utilization significantly.

Full duplex mode

In addition to the compatibility mode based on CSMA/CD, the Gigabit alliance will also standardize a collision-free full duplex mode which will offer better performance than the half duplex CSMA/CD mode. The only firm requirement will be support for the 1500 byte MTU size of CSMA/CD Ethernet. The arbitration algorithm and buffering policy of the full duplex mode is left to the actual implementations. Important elements will be buffering policy and link-layer flow control to ensure efficient link-layer data transfer without buffer overflow. The full duplex mode will be less complicated than the half-duplex mode and will probably be targeted for high-end end-systems and switch interconnection.

2.2 Deployment of Gigabit Ethernet

The introduction of high-speed networking technologies like Fast Ethernet introduces a new demand on local area backbones. The aggregated bandwidth will increase significantly and will require backbone upgrades due to all the shared networked resources (i.e. file server, WWW servers, printers). The Fast Ethernet switches today offer an internal Gigabit switching capacity, but interconnecting these devices in larger configurations requires links with Gigabit capacity. Figure 4 illustrates some of the possible configurations using Gigabit Ethernet. Gigabit Ethernet will be used as network access to file servers when the aggregated bandwidth of Fast Ethernet clients demand Gigabit bandwidth. Interconnection of Gigabit switches is another important area. This will be necessary when large clusters of end-systems using Fast Ethernet are interconnected. At the moment, Gigabit Ethernet will not be used to the desktop. However, the Gigabit Ethernet alliance have standardized a network technology which they think will offer acceptable network performance to the next generation of high-end computers which can take advantage of the Gigabit Ethernet speed.

2.3 Performance issues

In the last decade a lot of evaluations of the use of Internet technology (e.g. TCP, UDP, IP) over Ethernets have been done (see [7] and its references). Since Gigabit Ethernet is backward compatible with Ethernet, some of the conclusions in these papers are valid for Gigabit Ethernet as well. The difference is of course the utilization issue due to the extended Carrier Event. Another issue is the increased bandwidth and the effect it has due to the dynamics of for instance TCP, which is the transport layer protocol in Internet.

One problem for end-systems using Gigabit Ethernet is that most existing protocol implementations are not efficient enough to take advantage of the Gigabit bandwidth. The computer vendors are of course working on new and improved solutions based on techniques such as no-copy implementations or single copy implementations [8] to speed up protocol processing. This will be absolutely necessary to take advantage of the bandwidth available in Gigabit Ethernet.

The single most important advantage of Gigabit Ethernet is that the technology is well known and low cost compared to new technologies like ATM. In an Internet setting, Gigabit Ethernet should be a plug-and-play technology due to the backward compatibility with Ethernet.

One potential problem with Gigabit Ethernet is the CSMA/CD access method which is vital for the backward compatibility. The solution used in Gigabit Ethernet is an extended carrier and packet bursting. The extended carrier makes it possible to maintain a reasonable network diameter. The packet bursting makes sense as long as the sending entity can send several packets within a Carrier Event. The length of a Carrier Event is of course less than the 1500 byte maximum frame size. The overall impact of the extended carrier will depend on the traffic in the actual networks, but unless the packet bursting increases the utilization of the network, the small packets which are an integral part of typical Internet traffic will reduce the aggregated utilization of the Gigabit network significantly.

Figure 5 shows a contour plot of the network utilization (utilization 0.2 – 0.9) as a function of the average packet size for small packets, i.e. packets less than 512 byte, and the overall percentage of small packets. The model we have used is given by

$$\textit{utilization} = [1500(100 - x) + yx] / [1500(100 - x) + 512x]$$

where x is the percentage of small packets (i.e. less than 512 byte) and y is the average size of these packets.

This model does not consider the preamble and other overhead bytes, and it assumes that packets are sent back-to-back fully utilizing the link. In addition, we assume that all packets larger than 512 byte are MTU-sized, and that packet bursting cannot be used. The former assumption will increase the utilization, while the latter assumption will lower the actual utilization. Packet bursting will work better for aggregated traffic than for point-to-point connections. Yet it requires that the Gigabit Ethernet can send multiple packets within a time slot given by (1500 * 8) bits / 1 Gbit/s which equals 12 µs. As Figure 5 illustrates, the utilization increases as expected with an increase in the average packet size of small packets and a decrease in the small packet percentage. With 30 % small packets and an average small packet size of 96 bytes this gives maximum a 0.9 utilization.

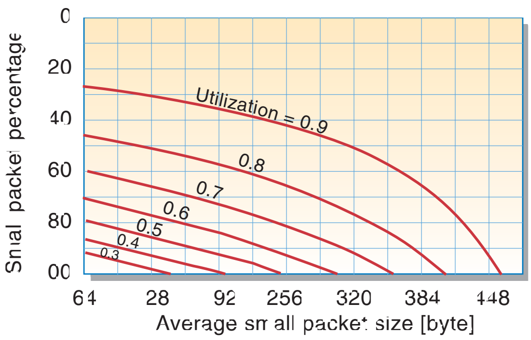


Figure 5 Gigabit Ethernet utilization

Another problem for servers using Gigabit Ethernet is the limited frame size. The frame size is necessary for backward compatibility. However, the per-packet cost of protocol processing has not decreased at the same rate as the per-byte cost over the last years. The impact of the frame size is illustrated in Table 1, which includes a set of ATM performance

Table 1 Performance of TCP/IP over ATM

Protocol	MTU size	Throughput
Classical IP over ATM	9180 bytes	102 Mbit/s
ATM LAN Emulation	1516 bytes	75 Mbit/s
ATM LAN Emulation	4544 bytes	93 Mbit/s
ATM LAN Emulation	9234 bytes	101 Mbit/s
ATM LAN Emulation	18 kbytes	85 Mbit/s

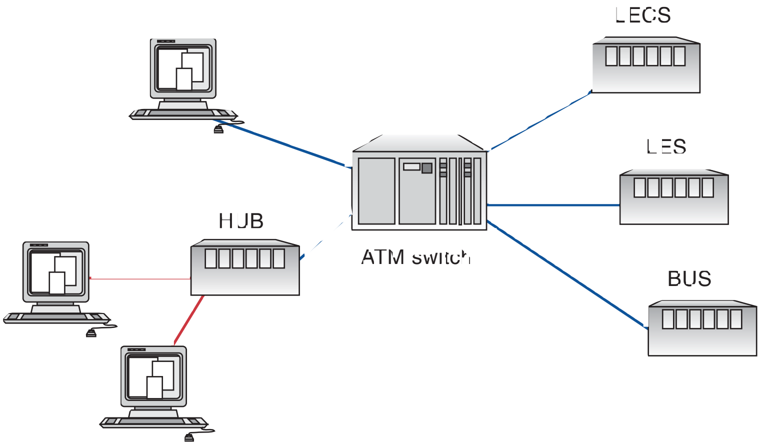


Figure 6 LAN Emulation components

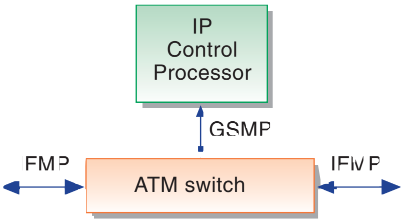


Figure 7 IP switching architecture

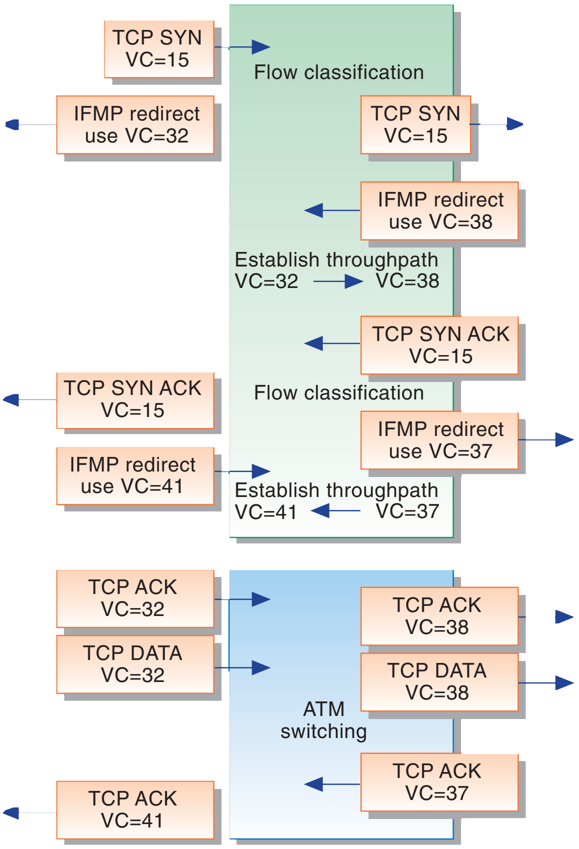


Figure 8 IP switch operation

which will be discussed in more detail in the next section. The maximum frame size affects the achievable throughput significantly.

3 ATM and Internet technology

Asynchronous Transfer Mode is a fast packet switching method originally developed for the public broadband integrated services network (BISDN) but has also been adopted for private high-speed networks both in the local and wide area. ATM is primarily being used in backbone infrastructures of large corporate networks and also by wide area network providers. Edge devices include routers and hubs with ATM interfaces and servers with ATM network adapters, as shown in Figure 10. ATM solutions for the desktop based on 25 Mbit/s links are also being deployed. In connection with Internet technologies, ATM is currently used for LAN emulation and Classical IP over ATM. Other solutions, most notably IP/tag switching, routing over large clouds (NHRP) and MPOA, are emerging.

Classical IP over ATM

Classical IP and ARP over ATM [9] is a solution to overlay an IP network on top of logical IP subnetworks (LIS) based on ATM. The solution is based on the use of several redundant ARP servers which offer ATM address resolution, i.e. map IP address to ATM NSAP address, within a LIS. LLC/SNAP encapsulation is used to demultiplex different higher layer protocols in the receiving end-systems. The default Maximum Transmission Unit (MTU) used by Classical IP is set to 9180 bytes which is roughly 6 times the MTU used by the Gigabit Ethernet. The IP protocol datagrams are mapped to a corresponding VPI/VCI combination through an extended ARP table. Hence, Classical IP will only set up a single VC between each pair of end-systems which will deny any QoS guarantees to the end-systems. Another problem with this approach is that direct ATM connection can only be established between hosts within a LIS.

LAN Emulation

LAN Emulation [10] is a short term solution which has been introduced to accelerate the local area deployment of ATM. The service offered by LAN Emulation is a bridged LAN on top of an ATM network. This is accomplished by offering a service interface identical to the interface in legacy LANs. Thus, ATM LAN Emulation works at the MAC layer in contrast to Classical IP over ATM which works on the Network (IP) layer. To emulate a traditional LAN environment (see Figure 6), services like Broadcast are implemented by the Broadcast and Unknown Server (BUS). Address Resolution between MAC addresses and ATM addresses are performed by the LAN Emulation Server (LES). In addition, multiple emulated LANs can be maintained by a LAN Emulation Configuration Server (LECS). The advantage of LAN Emulation is that existing infrastructure, i.e. computers, protocols and networking software, could be used without upgrading the core network infrastructure including network interface cards (NIC). By using a set of address and broadcast servers, an ATM network will behave just like a legacy LAN. One difference, though, is that LAN Emulation can offer MTU sizes ranging from 1500 byte to 18 kbyte.

LAN Emulation has just recently emerged as an acceptable solution for interconnecting legacy LANs. Meanwhile, Fast Ethernet has become the preferred high-speed networking technology to the desktop. The fact that no end-systems can utilize the ATM technology fully has not made the task any easier for the ATM vendors and their product portfolio.

Label switching

Recently, packet forwarding based on label switching in conjunction with network layer routing has become popular. Several different vendors have introduced different solutions using this scheme. The label-switching technology is expected to increase the scalability and performance of packet forwarding while introducing more flexibility in the basic routing services. The labels can for instance be used to identify and separate traffic with specific QoS parameters. Other advantages are the simplified integration with cell switching technologies like ATM and the fact that information about the physical network topology is made available to the network layer routing.

One of the label-switching solutions which has been introduced recently to integrate ATM and IP routing is IP switching [11] from Ipsilon Network Inc. The reasoning behind IP switching is that the tremendous effort invested in ATM hardware making it a reliable, scalable technology offering high bandwidth and multicasting functionality should be reused in Internet. Thus, IP switching will use the core ATM switch in conjunction with the control software found in existing Internet technology. The flow concept in the next generation IP protocol (IPv6) will make it possible for IP switches to avoid unnecessary processing for certain traffic streams by short-cut routing the traffic through the ATM switch. This approach, which has been promoted by Ipsilon Networks, requires two protocols in order to function properly, Ipsilon Flow Management Protocol (IFMP) [12] and General Switch Management Protocol (GSMP) [13] as shown in Figure 7. The GSMP will make the IP switch controller able to configure the ATM switch. The IFMP will make it possible for IP switches to exchange information making the short cut routing possible.

Figure 8 shows the operation in the IP switch during a TCP connection set up. The TCP SYN packet is forwarded over the default control VC (number 15) to the IP Control Processor which does a flow classification. The control processor decides that this connection will probably live long enough to benefit from IP switching and issues an IFMP redirect to the upstream node (host or router) asking it to use VC = 32 for this connection. The SYN packet is forwarded to the downstream node (host or router) on the default VC = 15. An IFMP redirect is then received from the downstream node asking the IP switch to use VC = 38 for the same connection. The IP switch can now set up a through path (VC = 32 to VC = 38) which will bypass the IPCP. Similar operations will occur in the opposite direction when the TCP SYN ACK is received and forwarded. When the final TCP ACK in the three-way-handshake is received, there will be through paths in both directions for the traffic from this connection. The TCP packets will be switched in the ATM switch bypassing the IP Control Processor. When the connection is terminated, these through paths will be removed.

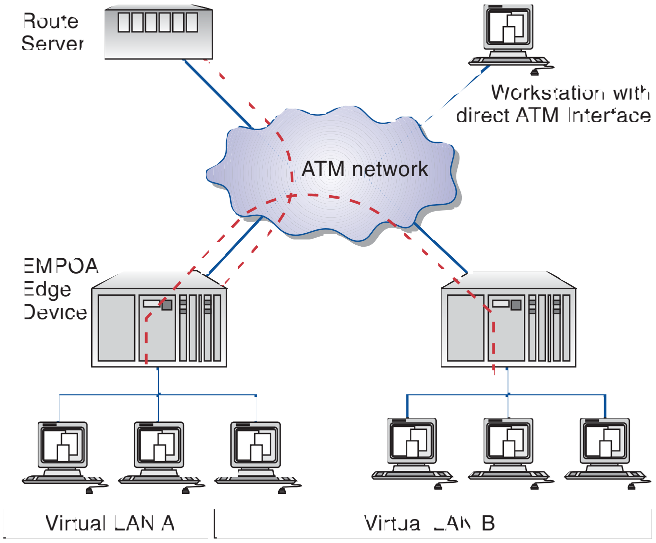


Figure 9 MultiProtocol Over ATM (MPOA)

Another label switching solution denoted Tag switching [14] is proposed by the leading router vendor Cisco. Tag switching is a more general architecture for efficient routing over different network technologies including the Ethernet network suite. The control and forwarding components are independent, and the tags are placed in the layer 2 or layer 3 header if the semantics of these layers allow it. Otherwise, the tag is inserted in a small shim header between the two layers. The tags should preferably be inserted at the edge of the network to make the routing more efficient through the entire network. The tags will either be distributed to or requested from neighboring Tag switches using a Tag Distribution Protocol (TDP).

Although a Tag switch can participate in routing protocol procedures, it can also provide forwarding over paths which are different from the paths determined by destination based routing. Load balancing is also possible due to the routing

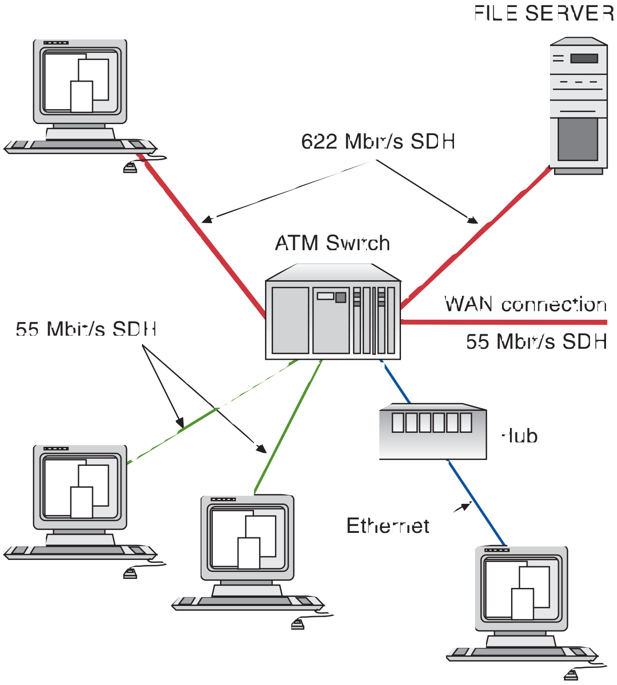


Figure 10 ATM deployment

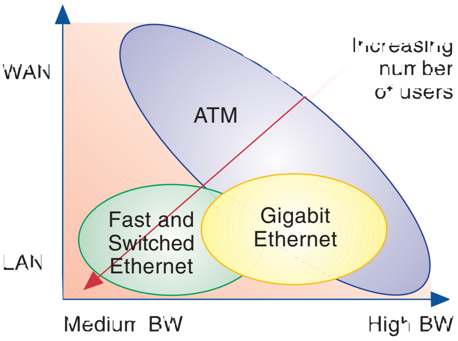


Figure 11 Market for ATM and Gigabit Ethernet

flexibility of the Tag switching scheme. Another advantage of the Tag switching scheme is that it can provide hierarchical routing, which decouples interior and exterior routing, by using a stack of tags. Thus, only the edge switches need to maintain exterior routing information. Tag switching can support QoS parameters provided that the necessary flow classification information is distributed in advance through a control protocol like RSVP. Optimization for specific network technologies like ATM is also possible in addition to seamless interworking with existing routers. Cisco has an advantage compared to the other label switching vendors due to its dominance in the router market.

None of the existing solutions fulfil all the requirements of an open standard capable of achieving all the benefits of the label switching scheme. Therefore, the Internet Engineering Task Force has established a working group to standardize Multiprotocol Label Switching (MPLS) [19]. The outcome of the standardization process is not yet available although the standard should be finalized later this year according to the preliminary timetable from the MPLS working group. Cisco which is responsible for the Tag switching proposal is an active member of the MPLS working group. The final standard will probably be similar to their proposal in many aspects. IP switching is so far only used for ATM networks and violates therefore one important MPLS requirement which is independence of the underlying network technology.

Next Hop Routing Protocol

The IETF is extending its support for IP over Non-Broadcast Multiple-Access (NBMA) networks like ATM. One key element in supporting IP over such technologies is the Next Hop Resolution Protocol (NHRP). NHRP makes it possible for end-systems connected to different logical IP subnetworks of the same NBMA network to communicate. This is not possible for LAN emulation and Classical IP over ATM where direct ATM communication can only be established between end-systems in a single emulated LAN or LIS. To make the short-cut ATM connection set up possible, NHRP uses an address resolution mechanism which works across multiple logical IP subnetworks between the sender and the receiver. The NHRP protocol will require the NBMA address of the receiver which makes it possible to establish a direct connection through the NBMA network bypassing the IP routing completely.

For NHRP to operate there must be a Next Hop Server in every LIS. This server will register all NBMA addresses and their corresponding IP addresses during the boot process of the end-systems.

A sending host will send a Resolution request to its NHS in order to resolve the NBMA address of the receiver. If the receiver is connected to another LIS, the NHS will forward the Resolution request to the next NHS along the IP routing path. When the request reaches the NHS in the LIS where the receiver resides, a Resolution reply is sent back to the sender via the routing path or the NBMA network. Intermediate NHS can cache registered address mappings to reduce the response time of subsequent requests.

MultiProtocol Over ATM (MPOA)

MPOA [15] is a recent standardization effort from ATM Forum, that encompasses both ATM LAN Emulation, Classical IP over ATM and the Next Hop Routing Protocol. The standard is heavily supported by a few ATM switch vendors, and represents a fierce competitor to the IP switching scheme. The MPOA model as shown in Figure 9 can be viewed as a "virtual router", where all edge devices can be regarded as router interfaces and the entire ATM network can be regarded as the router backplane.

When connections are set up between computers in two Virtual LANs, Edge Devices will establish a direct virtual circuit to the indicated destination. The mapping between remote MAC address and ATM address is based on information from a Route Server. Thus, packets are not forwarded to traditional routers, but transferred directly between edge devices on a dedicated Virtual Circuit.

3.1 Deployment of ATM

Currently, the installed base of ATM networks are mainly in LAN backbones and WAN interconnection. This is illustrated in Figure 10 where the desktop workstations are connected using 155 Mbit/s ATM and a file server utilizing a 622 Mbit/s ATM line. At the link level, SDH framing is used. ATM solutions utilizing even higher bandwidths (2.4 Gbps) are currently being implemented. To interconnect ATM and other networks (e.g. Ethernet, Fast Ethernet, Token Ring), routers or ATM LAN emulation are used.

The advantages of ATM are high bandwidth, support for Quality of Service (QoS) and its scalable architecture. For applications demanding high-bandwidth, ATM will outperform Gigabit Ethernet due to the larger MTU size. However, this will probably only be achieved in Intranets. In Internet, the next generation of IP will use Path MTU Discovery to find the largest possible MTU for a given path. Thus, if ATM is employed all the way, large MTUs can be used. If not, the smallest MTU available will be used. For Internet-type communication where many network providers are involved, it is not likely that all hops will use ATM, and such connections will be forced to use the smallest packet size supported by the intermediate networks. Another problem for use of ATM in Internet, is that most users will only be accessible via firewall routers due to the inherent security problems in the Internet. The current version of ATM signalling does not include any authentication procedures to avoid certain end-system attacks. Furthermore, most users rely on Ethernet or Fast Ethernet networks in local workstations and routers. The success of ATM in Internet will depend on the acceptance of ATM among the service providers.

3.2 Performance issues

Due to the larger MTU size, ATM has an advantage compared to Gigabit Ethernet for high-bandwidth communication, e.g. graphical industry, seismic interpretation, visualization and animation. For such applications, ATM can be the best choice also for use in workstations. The effect of the MTU size for a given platform can be seen by comparing the TCP throughput performance using ATM Lan Emulation and Classical IP over ATM for various MTU sizes.

The measurements presented in Table 1 are done by one of the authors using MacOS system 7.5 and Open Transport 1.1.1 on two Power Macintosh 9500/200 systems using Apple ATM software. A sequence of large buffers are transferred. For comparison, throughput for direct communication over AAL5 (avoiding the overhead of TCP/IP) is 130 Mbit/s. As we can see from the table, there is a high performance penalty for small MTU sizes. The best performance is obtained by Classical IP with MTU equal 9180 bytes. For LAN Emulation, TCP throughput increases with increasing MTU. However, for the highest MTU, 18 K, we could measure lower performance primarily due to buffer management overhead. For this protocol stack, an MTU size of 9180 bytes is optimal. In general, larger MTUs will give a lower CPU utilization and a higher throughput. This relationship tells us that Gigabit Ethernet will suffer from its fixed 1500 byte MTU, and require more processing power than ATM for the same throughput requirements.

4 Concluding remarks

Figure 11 shows how Gigabit Ethernet fits in between current technologies. It is anticipated that Gigabit Ethernet will have a higher volume than ATM as illustrated by the intensity of the red background colour, due to its inherent interoperability with the existing Ethernet technologies. This is particularly important in the local area, where investments in the existing infrastructure is highest. The Ethernet networking suite will get a technology which is capable of providing the necessary bandwidth to interconnect the rapidly increasing base of current Fast and Switched Ethernet networks, as well as high capacity server connections in local and corporate networks. For high speed networking in the local area, Gigabit Ethernet will greatly improve the performance of networks currently based on Fast and Switched Ethernet, while ATM is expected to be the preferred WAN technology for high capacity network interconnection.

ATM will play a major role in the WAN area and in large corporate backbone networks. We will see ATM interfaces in high speed hubs, switches and high end servers. In particular, ATM technologies that support current technologies like LAN Emulation and possibly MPOA or IP/Tag Switching will be incorporated in the network edge devices (such as hubs and routers). In addition, there will be a small market segment for ATM in specialized multimedia workstations that require the guarantees in Quality of Service that ATM supports. Such applications include conference rooms, seismic interpretation, telemedicine applications, etc. In addition, there will be a need for very high capacity network interfaces in the next generation information servers for corporate and public use. Such servers will support thousands of simultaneous connections, possibly with QoS guarantees. For this purpose, ATM has the appropriate functionality.

However, it is not only technical arguments that decide the battle over the next generation network infrastructure. Politics and strategies imposed by leading network operators and equipment vendors may be equally important. History shows that superior technologies may not always succeed.

Use of IP directly over SDH by means of PPP is another interesting network alternative to ATM networks. Several router vendors are already offering network modules with pure SDH interfaces. IPv6 can be mapped directly above PPP and SDH and the inherent 5 byte header overhead in ATM cells can be avoided. The idea is that SDH management should be able to reroute the traffic in case of problems and routers instead of the ATM network providers will get the benefit of statistical multiplexing. In addition, RSVP and IPv6 will be used to offer QoS to the users. The advantages are the reduced overhead. The drawbacks are the lack of possibility to police the traffic, and the lack of flexibility of QoS-based ATM VPI and VCI switching. In addition, the long distance network provider can no longer take advantage of statistical multiplexing. This scheme requires that the RSVP or similar functionality is available in the IP network to perform resource reservation. Another interesting technology which may be used for SDH transmission by wide area operators is Wave-Division Multiplexing. This technology is cheaper than setting up new fibre links. By multiplexing several different wavelengths over a single SM fibre, the overall link capacity will increase linearly with the number of different wavelengths.

So far, most existing and planned wide area network infrastructure is based on ATM. ATM is currently the preferred high-speed networking technology among wide area network providers. This resource commitment will be important when the Internet and Intranets are being upgraded. The most important factor is how cost effective each solution is for the network provider and how the functionality is tailored to the requirements of the end-users. One of the weak points of current ATM technology is the lack of basic security in the ATM network itself. Thus, most companies will force all traffic through a company firewall to prevent security attacks.

References

1 Stallings, W. IPv6 : The new Internet Protocol. *IEEE Communications,* 1996.

2 *RFC Resource Reservation Protocol.*

3 Atkinson, R. *IP security architecture.* Internet Engineering Task Force, 1995. (RFC 1825.)

4 Johnson, D B. *Mobility support in IPv6.* Internet Engineering Task Force, Internet draft (work in progress).

5 *Gigabit Ethernet alliance technical white paper.* See http://www.gigabit-ethernet.org.

6 Halsall, F. *Data communications : computer networks and open systems.* Addison/Wesley, 1995. ISBN 0-201-42293.

7 Moldeklev, K et al. TCP/IP behavior in a high-speed local ATM network environment. *Proc. of the 19th IEEE conference on local computer networks,* Minneapolis, Oct. 1994, 176–185. ISBN 0-8186-6680-3.

8 Ahlgren, B et al. The performance of a no-copy API for communication. *Third IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPCS'95),* Mystic, Connecticut, August 23–25, 1995.

9 Laubach, M. *Classical IP and ARP over ATM*. Internet Engineering Task Force, 1995. (RFC 1577.)

10 ATM Forum. *LAN emulation over ATM v1.0 Specification,* January 1995.

11 Newman, P et al. Flow labelled IP : a connectionless approach to ATM. *Proc. of IEEE Infocomm,* San Francisco, March 1996.

12 Newman, P et al. *Ipsilon flow management protocol specification for IPv4.* Internet Engineering Task Force, 1996. (RFC 1953.)

13 Newman, P et al. *General switch management protocol specification.* Internet Engineering Task Force, 1996. (RFC 1987.)

14 Cisco, http://www.cisco.com/

15 *MPOA version 1.0.* ATM Forum, 1997. See http://www.atmforum.com

16 Sachs, M W, Varma, A. Fibre Channel and Related Standards. *IEEE Communication Magazine,* 34, (8), 1996, 40–49.

17 Widmer, A X, Franaszek, P A. A DC-Balanced Partitioned-Block 8B/10B Transmission Code. *IBM Journal of Research and Development,* 27, (5), 1983, 440–451.

18 ATM Forum. *User-Network Interface Specification, Version 3.1.* Prentice-Hall, September 1994.

Haakon Bryhni is PhD student at the University of Oslo, Department of Informatics, supported by the Norwegian Research Council. His research interests include high capacity communication systems, network architecture, protocols and computer architecture for multimedia communication. In particular, interconnection of high capacity servers and new demands placed on the network subsystem to support high capacity communication streams. Bryhni holds an MSc from the Norwegian Institute of Technology in Trondheim, and has broad experience with communication systems from industry (Apple, Alcatel, Telenor), military (NDRE) and consulting.

email address:
bryhni@ifi.uio.no
URL:
http://www.ifi.uio.no/~bryhni

For a presentation of
Espen Klovning, see page 28.

Predicting network server performance

ESPEN KLOVNING AND ØIVIND KURE

1 Background

In this article we explore the impact increases in processing capacity and memory speed will have on protocol throughput in a general purpose workstation. The memory bandwidth and processing capacity are well known bottlenecks, and there are ongoing research efforts aimed at protocol architectures and implementations that minimize the effect of these bottlenecks. In this article we take a different approach; assuming a protocol implementation similar to the existing one, we explore the impact increases in processing capacity and memory speed will have on protocol throughput. This has a direct bearing on the suitability and more important it is a forecast of the timing when general purpose workstations can be used as efficient servers in the network.

Our interest is not peak bandwidth per se, but the resources required to run multiple streams with moderate frame lengths. In order to do so we built a simulation model of the protocol processing in a general purpose workstation. The full protocol stack is simulated, but retransmission is not included in the work load. Retransmission should be viewed as an external parameter, since it depends on the network itself and more important, the service subscribed to from the network operator.

The workload was modeled as 65 two Mbit/s TCP streams, i.e. roughly the number that can run on top of a 155 Mbit/s ATM connection. The peak rate of 2 Mbit/s is arbitrary, but is a reasonable approximation for the desired rate in a server for WWW, FTP, and Multi Media Content. The research was initiated as part of a study of Video-on-demand services based on general purpose workstations as video caches in the network. The protocol stack for such a cache will have to offer reliable services when the cache is loaded, and an unreliable and rate controlled stream to the different customers. The TCP/IP stack without any errors in the transmission is a reasonable approximation of such a stack; all data structure and management of error and flow control are included in the stack, while the more costly part of actual retransmissions is excluded. The results for TCP will at least be a first approximation.

Whenever a simulation model is used, the most important issue is the validity of the model. An additional reason to use the TCP/IP stack as model of the protocol stack is the availability of measurements for different machines. The simulator can then be validated through the large number of available measurements.

The throughput of the TCP/IP protocol stack is a function of many parameters. In a previous work [2] we have shown, through measurements, the large impact processor speed, window size, and user payload length will have on the throughput in a local area ATM network. There is no linear relationship between these factors and throughput. In addition, our analysis displayed the sensitivity in the performance to timing and detailed protocol behavior. The protocol performance in a network server can therefore not be extrapolated directly from existing performance measurements. Instead, the actual protocol behavior will have to be simulated.

The main contribution of this article is a prediction of protocol performance as a function of processor speed, memory speed, window size, and user payload

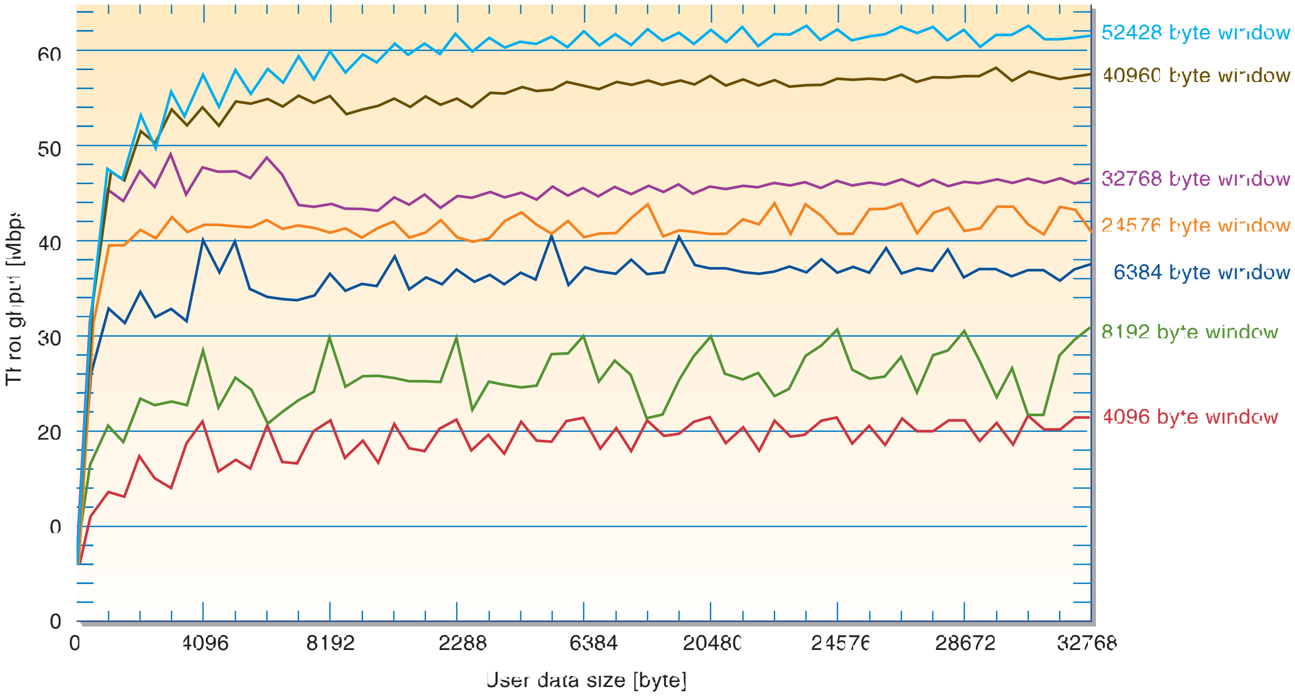


Figure 1.1 TCP throughput LCN'94

size. These results can then be used to judge timing and cost effectiveness of a general purpose workstation as servers in a network.

The article is outlined as follows. In the remainder of this section we identify factors that influence the performance of TCP. A detailed model of the protocol processing requires a detailed measurement for setting the parameters. As important is the validation of the model and the parameters used. All of this is described in Chapter 2. The results are discussed in Chapter 3, while a brief conclusion is presented in the last chapter.

1.1 Factors influencing TCP performance

The TCP protocol stack, and in particular its implementation for BSD UNIX derived operating systems, has been a continuous topic for analyses. These analyses consider networks with lower bandwidth or smaller frame transmission units than the cell-based ATM network can offer through the ATM adaption layers. In a previous paper [2] we discussed some of the important factors and their effect on the TCP performance in a local area ATM network.

Figure 1.1 shows the memory-to-memory throughput measured for different user data sizes and different socket buffer sizes. The software parameters with the largest influence are the maximum window size and the user data size. In general, the throughput increases with increasing window and user data size. However, it is not a monotonic behavior, the graphs have their peeks and drops. These fluctuations in throughput are caused by more subtle interaction between the protocol and the operating system including the socket layer. This interaction influences the behavior of the protocol and makes it difficult to predict the packet flow based on the system parameters.

TCP is byte stream oriented. Thus, the segmentation of the byte stream depends on the user data size, the window flow control, the acknowledgment scheme, Nagle's algorithm [18] which prevents transmission of too many small segments, and the interaction between the operating system and the TCP implementation.

The functionality and speed of the host and network interfaces also influence the performance; more powerful machines and more advanced interfaces can affect the timing relationships between data segments and window updates and acknowledgments. A window update is usually sent if the highest announced window sequence number edge can slide more than 2 times the MSS or at least 35 % of the receive window. TCP uses either immediate acknowledgments or delayed acknowledgment piggybacked on either data segments or window updates. In addition, acknowledgments are generated periodically every 200 ms. The reception of an acknowledgment may therefore trigger transmissions of new segments based on the current number of bytes in the socket send buffer. The timer generated acknowledgments occur asynchronously with other connection activities. Therefore, such an acknowledgment may not adhere to the window update rules above. As a consequence, a timer generated acknowledgment can change the segment flow on the connection.

2 Method

A simulator was designed and implemented to predict the performance of the protocol stack as a function of end-system performance. In this section we describe the model, the parameters in the model, and how they were measured. The final subsection contains the validation of the model.

2.1 Model

The free variable in our simulation is the performance of the end-system. The intention of the study is to predict the protocol performance over a wide range of end-systems. The free variables are translated into resources used by the protocol stack. A minimal abstraction is to model the end-system by two resources, CPU speed and memory bandwidth. In addition, the processing of core operating system (OS) functionality was included as a resource, since existing research [6,7] has shown that some basic operating system functions do not scale with CPU speed and memory bandwidth. Typical examples are system calls and sleep/wakeup functionality.

The actual protocol performance will be a function of all the hardware and software implementation details in the end-system. The most significant missing element from our abstraction is the cache. The results would have been more accurate with the cache included, but the disadvantage would have been a restriction of the results to a specific memory architecture. In addition, future hypothetical end-systems are part of our study, and obviously no assumption about the detailed memory architecture can be made.

Overall, we feel that the abstraction of an end-system into three resources, CPU, memory bandwidth, and operating system is sufficient for the purpose of the study. A model with a finer granularity would have required more detailed modeling of the CPU, memory system, and operating system. This would not have improved the results, since few assumptions can be made about the architecture of future end-systems.

2.1.1 Design and implementation overview

The event-driven simulator is implemented in C and mimics the communication system in SunOS 4.1.3, including TCP/IP and the socket layer implementation. The internal hardware architecture is modeled as system resources, while user data, protocol code, and ATM cells are implemented as finite state machine objects. These objects process at different priority levels corresponding to the priority levels used in SunOS 4.1.3 and similar BSD based UNIX versions [9]. The simulator can be used to simulate the throughput of distributing *n* concurrent traffic streams from a single server to *n* receivers. Simulating end-to-end throughput between two end-systems is a special case.

In our simulations, we have used the SPECint92 numbers as an indication of the processing capacity of the particular end-system processor. A similar prediction method is presented in [13] where performance predictions of a designed network interface is the issue. The relative memory speed can be deduced from the bus width and the cycle time, or via benchmarking the memory system.

2.2 Tracing

There are several different approaches which can be used to quantify the processing cost of different parts of the communication system. It is possible to manually count instructions, to trace the CPU

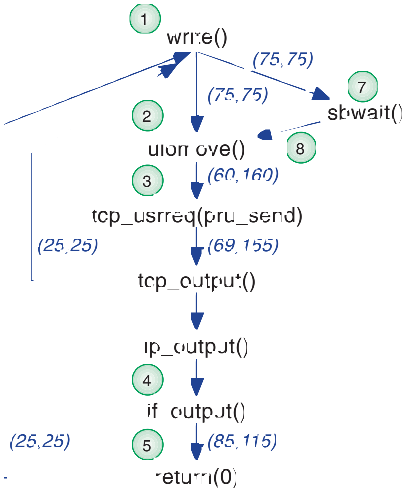


Figure 2.1 The main send processing paths. The processing cost of the different parts of the protocol stack is shown in parenthesis for 1 and 4 kbyte packets

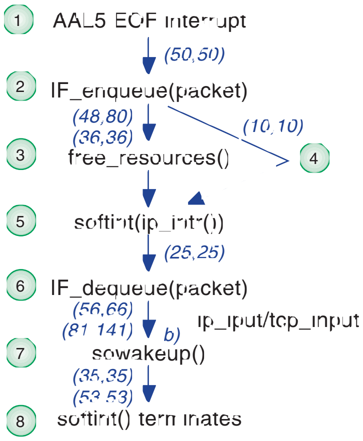


Figure 2.2 The main receive processing paths in both the sender and the receiver. The processing cost of the different parts of the protocol stack is shown in parenthesis

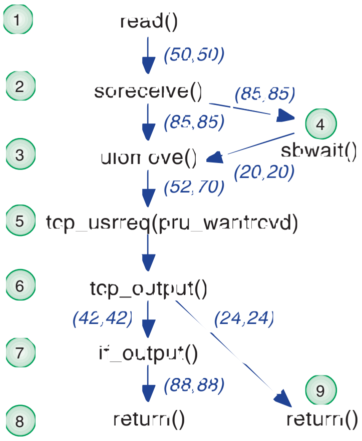


Figure 2.3 The copy out process in the receiver. The processing cost of the different parts of the protocol stack is shown in parenthesis for 1 and 4 kbyte packets

instruction stream with a logic analyzer hooked on to the memory bus, or to instrument the communication system. Newer processors (e.g. UltraSparc, Pentium, Alpha) include a set of performance monitoring counters which can monitor important low-level events (e.g. cycles, instructions, cache hit and misses) in the computer. We chose to instrument the communication system due to previous experience with this technique.

Nearly all parts of the communication system in our reference workstation were instrumented with small probes, which were used to monitor the communication system while running memory-to-memory tests with the ttcp benchmark program. The reference workstation was the Axil 311/51 Sparc 10 clone, which is equipped with a 50 MHz TI SuperSparc processor with a 1 Mbyte external cache. The operating system was SunOS 4.1.3. Axil 311/51 has a 64 bit cache coherent memory bus (MBus[7]) and a separate 32 bit I/O bus (SBus[8]) which is accessed via an MBus-to-SBus (M2S) bridge. The SPECint92 value of this workstation is 65.2. The ATM network adapter was the SBA-200 network adapter from FORE Systems Inc. which uses DMA in both the send and receive path, and utilizes the burst size on SBus. The probes recorded the event information (i.e. event code, µs time stamp, and an integer value which usually was the packet length) in a large table pinned in physical memory. The overhead of the probes has been measured to be approximately 5 µsec. All presented timing results are in µs and are already adjusted for the probe overhead.

The main purpose of the instrumentation was to find the processing cost of different operations in the protocol stack processing, and to identify which category they belong to. In the OS class, we have included the time it takes to initiate and return from all write() and read() system calls. The low level interrupt processing is also part of the OS class. In the MEM class, we have included all operations which have a per-byte cost. This includes the TCP/IP input and out processing, the uiomove operations copying data across the user/system space boundary, and the device driver enqueue and deallocation operations. All the other per-packet costs have been included in the CPU class.

The main instrumentation of the communication system is depicted in Figure 2.1, Figure 2.2, and Figure 2.3. The first figure shows the entire call graph of frequently used send paths from the write() system call until the packet is queued for transmission in the ATM Adaptation Layer interface [15][1]. Readers unfamiliar with the terminology used in these figures can find more information in [9] and [16].

There are two paths which do not result in immediate transmission of an AAL5 frame. The first delayed path will be taken when the send socket buffer is full and the sending process goes to sleep (sbwait()) on the socket. Its operation will be resumed when TCP issues a *sowwakeup* call due to release of buffer space in the send socket buffer. This will happen when an acknowledgement from the receiver indicates successful transmission of data. The other delayed path will be taken if Nagle's algorithm [18] prohibits the protocol from sending data. In this case, the tcp_output() call will return prematurely to the write system() call. This algorithm makes the send path unpredictable since tcp_output() might be deferred until a large packet can be sent.

The measurements of the send path are presented in Table 1. As illustrated in Figure 2.1, we have combined the processing cost of TCP and IP processing even though the memory-bounded checksum calculation is part of TCP. The reason we did this was that other parts of TCP also depend on the data length, and that we wanted to keep the number of different internal parameters in our simulator as low as possible. Instead, we

[1] *Internal interface between frame processing (device driver) and cell handling (adapter firmware).*

Table 1 Timing of the different functions in the send path (sender) displayed in Figure 2.1

Size in Bytes Path	1K	4K	8K	9148	Classification
S1,2	75	75	75	75	OS
S1,7	75	75	75	75	OS
S2,3	60	160	NA	NA	CPU,MEM
S3,4	69	155	260	294	CPU,MEM
S4,5	85	115	NA*	175	CPU,MEM
S5,6	25	25	25	25	OS

* *In our measurement setting Nagle's algorithm prevented transmission of this packet size.*

approximate the per-byte cost of the checksum calculation and the buffer operations with the per-byte processing cost of the combined TCP and IP processing.

All results are named according to the numbering in Figure 2.1. For example, S1,2 is the time from the write() is issued until the uiomove process is about to start, while S3,4 includes the time from the tcp_usrreq*(PRU_SEND)* call until the if_output() device driver routine is called. The columns in Table 1 show the processing cost in µs for 1024, 4096, 8192 and 9148 bytes of user data. The latter value is the maximum MSS of an ATM network using VC multiplexing, while 4096 bytes is the maximum chunk of data copied from user space to the send socket buffer.

In addition to the send paths depicted in Figure 2.1, an acknowledgment from the receiver may initiate immediate transmission of a packet from the sender. This will happen if the acknowledgment indicates that there is no outstanding data. If so, TCP and Nagle's algorithm will allow transmission of any unsent data in the send socket buffer. Due to Nagle's algorithm, there will be less than *MIN(MSS, half the window size)* bytes in the send socket buffer. The acknowledgment processing, i.e. tcp_input(), will call tcp_output() which will transmit the packet. The immediate packet transmissions will follow the main send path which is illustrated in Figure 2.1. The only difference is that after the tcp_output() routine returns, the tcp_input() processing will continue.

There are two parts of the protocol stack processing of the send path where we have had difficulties finding the processing cost. The first part is the device driver processing. The main operation in the send part of the device driver is to enqueue the supplied packet for transmission. Since the network adapter uses DMA to transfer the packet from main memory to the network adapter, the output routine of the device driver can return after the packet is enqueued. Enqueuing packets for DMA transfer should be a rather straightforward process. However, in our instrumentation we have measured a bimodal device driver processing cost. The device driver processing cost for the first packet which is sent after an acknowledgment has been received is nearly half of the cost of enqueuing subsequent packets. For MTU-sized packets, the device driver processing costs are 175 µs for the first packet and more than 350 µs for subsequent packets. Similar behavior has been reported for the Fore ATM adapter for Apple Macintosh too. We decided not to pursue the cause of this problem any further.

Another part of the protocol stack processing which has been difficult to quantify is the time it takes to return to the uiomove() process after waking up a sleeping write socket. Usually it takes 20 µs, but it varies from 20 µs to nearly 360 µs which makes the mean processing time less valuable. We have instrumented all parts of the communication system but have failed to identify the factors causing the large variations. In the simulator we used the most frequent value which is 20 µs. Some of the processing costs we have presented in Table 1 have been classified as both CPU and MEM dependent, i.e. both per-packet and per-byte costs. Linear interpolation was used to separate these factors.

The main path of the asynchronous receive processing, from the interrupt to the socket wakeup call, is depicted in Figure 2.2. Since the receive processing in the sender (S)[2] and the receiver (R) are very similar, the results from the instrumentation is therefore presented in the same table. For the parts which differ, we have included the timing results for both cases. Table entries valid for both workstations are denoted B, while the entries which are only valid for the sender and the receiver are denoted S and R respectively.

As indicated in Figure 2.2, some transmit resources are deallocated during the interrupt processing. These limited resources (*mbufs* [9] and IOMMU address mappings [7]) have been used during the packet transmission, and must be deallocated after the entire packet has been transmitted. Since the device driver returns after enqueuing a packet, there are only two applicable solutions for releasing these resources. The network adapter could either interrupt the host and ask for resource deallocation, or the deallocation could be done during the send or receive driver processing of subsequent packets. The device driver we have modeled use the latter solution, and release resources immediately after the PDUs have been handled but before the receive interrupt terminates. The bimodal device driver processing we discussed earlier could be related to the transmit resource pool.

The amount of resources which are allocated depends on the length of the enqueued packet. Thus, in our simulator where the data transfer is unidirectional, these transmit resources are only needed in the sender. In the receiver, such transmit resources are only needed when acknowledgments are transmitted. Normally, a receive interrupt in the receiver would only free the resources corresponding to transmission of a single acknowledgment. This is reflected in

[2] *The sender (i.e. server) receives acknowledgements and window updates.*

Table 2 which illustrate that the processing cost (R2,3,5) in the receiver seems to be byte-independent, while the processing cost (S2,3,5) in the sender depends clearly on the byte count.

The paths through the TCP and IP input processing are different in the two workstations. In the receiver, checksummed incoming user data is appended to the socket receive buffer and the receiving socket is woken up. In the sender, the received PDU is either a window update with a piggybacked acknowledgment or a timer generated acknowledgment. For simplicity, we refer to both types as acknowledgments. Usually, some user data will be acknowledged releasing buffers from the send socket buffer. Thus, both these input processing paths are byte-dependent as illustrated in Table 2.

The process of copying user data out to user space is shown in Figure 2.3. The soreceive() routine which is called from the read() system call will either start copying user data to user space immediately, i.e. uiomove(), or issue a sbwait() if there is no available data in the socket receive buffer. In the latter case, the uiomove() process will be restarted when user data has been appended to the socket receive buffer. After the uiomove process is done, that is, the socket buffer is empty or the user buffer is full, the soreceive() routine will call tcp_usrreq*(PRU_WANTRCVD)* to update the window information and possibly send an acknowledgment. Tcp_output() will transmit a window update if the window can slide sufficiently. Thus, the tcp_usrreq() might result in transmission of an acknowledgment.

The uiomove process in the receiver is more efficient than the uiomove process in the sender. In the sender, the uiomove process is done on 1 kbyte chunks of data. In the receiver, the ATM device driver uses 8 kbyte large loaned buffers to store the incoming user data. Due to the use of loaned mbufs, the uiomove() process is more efficient in the receiver than in the sender. If a window update is required, the resulting TCP and IP processing is byte-independent. This is also the case for the device driver processing given that nothing but 40-byte long acknowledgments are transmitted. If no window update is sent, the tcp_output() call, and in turn the read system call, will return.

Table 2 Timing of different functions in the receive path (sender) displayed in Figure 2.2

Size in Byte sender Path in Figure 2.2	1K	4K	8K	9148	Classification
S2,3,5*	48	80	125	145	CPU, MEM
S6,7	56	66	85	115	CPU, MEM
S7,8	35	35	35	35	OS
Byte receiver	1K	4K	8K	9148	
R2,3,5	36	36	36	36	CPU, MEM
R6,7	81	141	230	262	CPU, MEM
R7,8	53	53	53	53	OS
B1,2	50	50	50	50	OS
B2,4,5	10	10	10	10	OS
B5,6	25	25	25	25	CPU

* *Deallocating 1K, 4K, 16K, 32K respectively.*

Table 3 Timing of the copy out process (receiver) displayed in Figure 2.3

Size in Byte Path in Figure 2.3	1K	4K	8K	16K	Classification
R1,2,3	85	85	85	85	OS
R1,2,4	85	85	85	85	OS
R4,3	20	20	20	20	OS
R3,5	52	70	100		CPU, MEM
R5,6,7	42	42	42	42	CPU
R7,8	88	88	88	88	CPU
R5,6,9	24	24	24	24	CPU
R9,1	14	14	14	14	OS
R8,1	16	16	16	16	OS

2.3 Validation

Figure 2.4 shows both the measured and simulated throughput as a function of the user data size for two different TCP window sizes, i.e. 8 kbyte and 48 kbyte. The measurements are denoted *M 8192 byte window* and *M 49152 byte window* in the figure legends. The corresponding simulations are denoted *S 8192 byte window* and *S 49152 byte window*. The simulations were done for an ATM network with 140 Mbps TAXI physical interface, i.e. minimum 3.14 µs per cell (53 + 2 bytes). As illustrated, the match between the measurements and the simulations are reasonably good for both window sizes and the studied range of user data sizes. However, the simulated throughput for the largest window size is a bit higher which we attribute to the neglected bimodal distribution of the ATM device driver processing times found during system profiling. For the smallest window size, the simulated throughput follows the throughput fluctu-

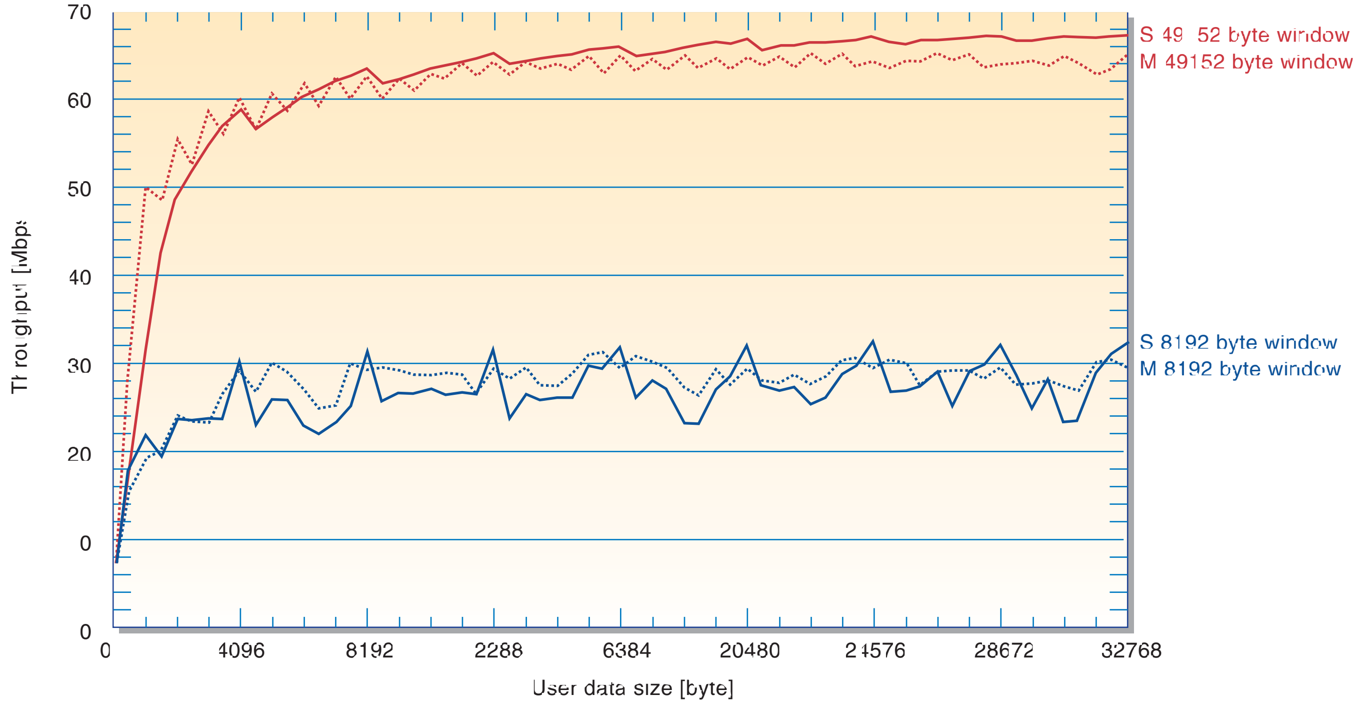


Figure 2.4 Measured and simulated throughput for Sparc 10 clone

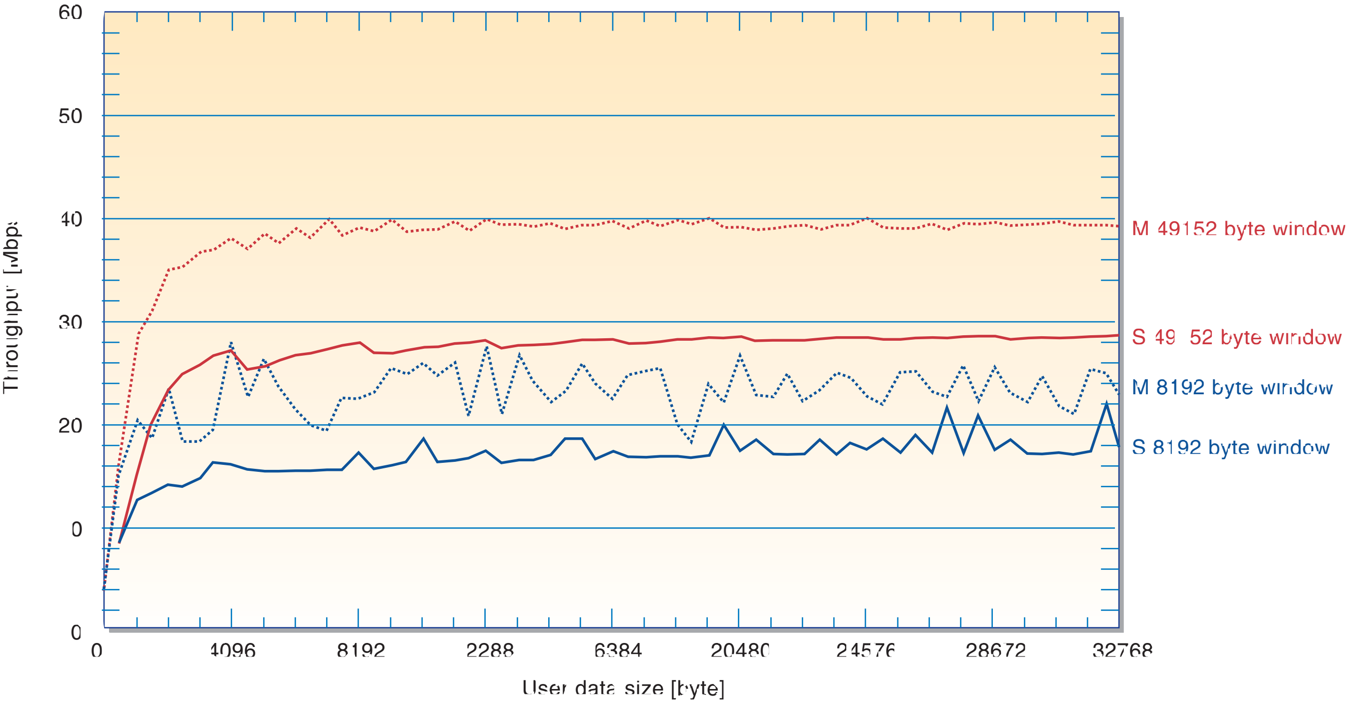


Figure 2.5 Measured and simulated throughput for Sparc IPX

ations of the corresponding measurements reasonably well, even though the variation is smaller. A smaller variation is not unexpected since the throughput for small window sizes is more sensitive to the correctly estimated processing costs of the different events than larger window sizes.

In order to validate our simulator, we also simulated the throughput performance between two older Sun SPARC IPX (4/50) workstations with a 21.8 SPECint'92 value, and compared these simulations with actual measurements for the same ATM physical interface, i.e. 140 Mbps TAXI. The measurements were done between workstations running an older operating system (SunOS 4.1.1). Based on benchmark tests, we found the relative memory speed to be 2.2 and the OS factor to be 1.2 [19].

The measured and simulated throughput as a function of user data sizes are shown in Figure 2.5. Figure 2.5 illustrates that the simulated throughput of the Sparc IPX workstation differs from the actual measurements in two ways. First, the simulated throughput level is significantly lower (25 %) than the measured throughput level. Second, the variations in the simulated throughput for the smallest window size is much lower than the variation in the corresponding measurements for the same window size. We attribute the latter difference to the sensitivity of the timing of different events in the protocol stacks. It is clear that for small window sizes, where the number of outstanding data bytes are small, the throughput will depend heavily on the timing of the different events.

However, the significantly lower simulated throughput is caused by differences in the internal architecture of the reference workstation and the IPX. In the IPX, SBus is used as a combined memory and I/O bus, which makes access to the network adapter from the CPU much faster. Thus, the device driver processing of both the send and receiver path will not scale according to the SPECint92. Hence, our simulator cannot be used to simulate the throughput performance of the IPX workstation accurately. In [2], it is shown that the device driver processing on the Sparc IPX is actually faster than on the Axil 311 workstation.

In some additional simulations, we assumed that the part of the protocol stack processing which accesses the network adapter, i.e. device driver processing, is equally fast for the Sparc IPX and the reference workstation. With this assumption, the maximum simulated throughput is within 10 % of the measured throughput. Simulating the throughput with more accurate processing costs of the device driver access of the IPX would probably reduce the difference even more.

To summarize, it is very difficult to simulate performance of general purpose workstations since the internal architecture is different. But based on our validation, we feel that our simulator is reasonably accurate when it comes to simulating end-system performance and end-to-end throughput. Problems related to different bus architectures should be considered when the simulator is used. Fortunately, use of a combined memory and I/O is not common anymore. Most of the results in the next section are from simulations of multiple simultaneous connections. Since our simulator has not tried to model the operating system in every detail, the results presented in this article will be optimistic due to coarse grained modeling of the interaction between multiple streams.

3 Results

The main goal with the simulator is to predict performance of a particular end-system distributing multiple 2 Mbps TCP streams.

3.1 Multiple streams – standard implementation

Figure 3.1 shows a 3D plot of the average throughput for 65[3] concurrent streams distributed from a single end-system connected to a 155 Mbps ATM link. The simulations were only done for OS factors 0.25, 0.5, 0.75 and 1.0, and memory factor 0.2 to 1.0 with 0.1 increment. These factors indicate the relative performance compared to the reference workstation, i.e. 0.5 means twice as fast and 0.25 means four times as fast. The window size was set to 64 kbyte and the user data size was 16 kbyte. In these simulations, the receiving end-systems had the same configuration as the distributing server. This will probably not

[3] *We chose 65 streams since the physical link should be able to carry 65 * 2 Mbps = 130 Mbps of user data.*

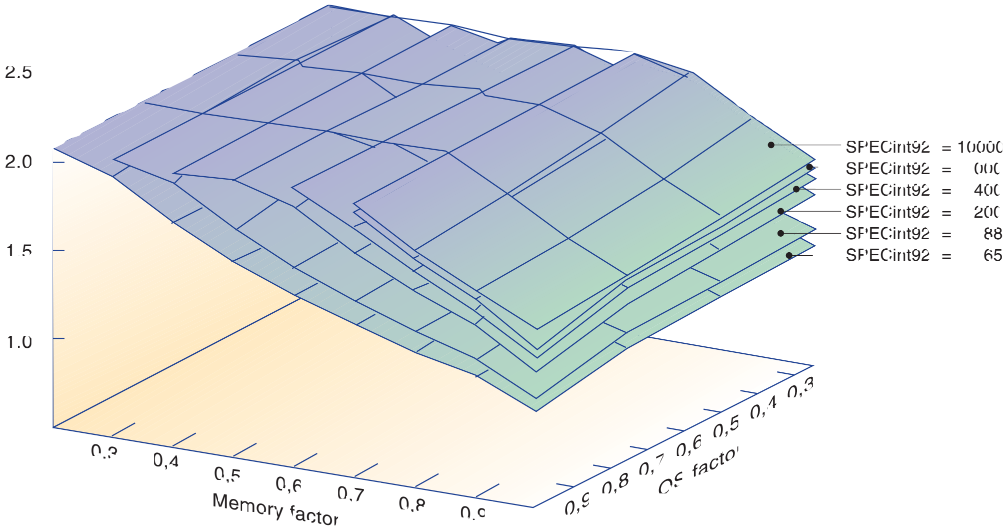


Figure 3.1 Average throughput of multiple streams

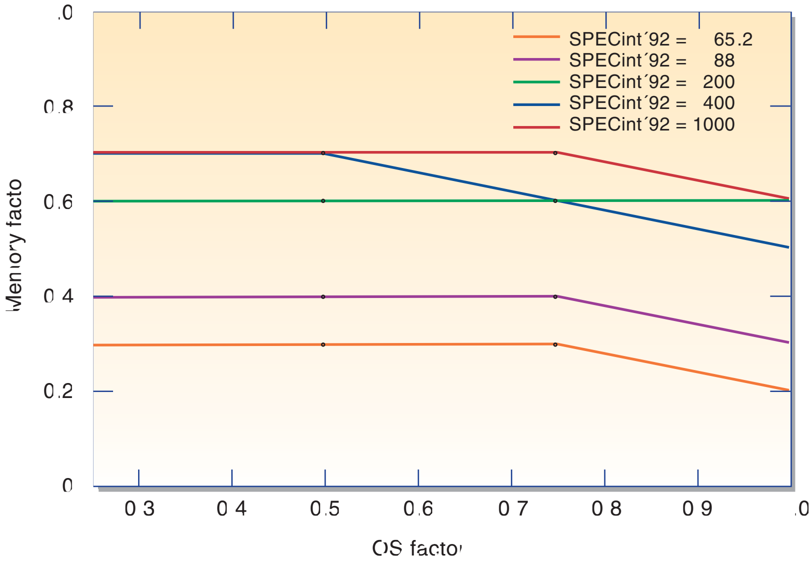


Figure 3.2 Contour lines at 2 Mbps average throughput for the different SPECint'92 values

be the case in a real network, but the load on the receiving end-systems is very low. Thus, using the server configuration during the simulations should not distort the results significantly. The x-axis is the memory factor and the y-axis is the OS factor. The z-axis shows the average throughput in Mbps.

In these simulations, 65 concurrent streams can maximally reach an average of 2.05 Mbps. The 3D plot in Figure 3.1 shows the average throughput for different SPECint'92 values between 65.2 and 10000. It is obvious that increasing the CPU capacity will improve the performance of this server. However, as the SPECint'92 values get higher, the overall improvement due to the increase in CPU capacity decreases. The operating system factor does not influence the overall throughput performance that much, which was expected. It is however illustrated clearly that the memory speed is very important for the overall throughput performance.

Figure 3.2 shows a contour plot for the average stream throughput for the five lowest simulated SPECint'92 values. The contour lines show the contours for an average throughput of 2 Mbps. Due to the coarse granularity of the simulation parameters, the contour lines for some of the different SPECint'92 values coincide for some settings.

It should be clear that medium-range end-systems such as the Axil 311 cannot distribute 65 streams with 2 Mbps average throughput unless significant reductions in memory speed and OS functionality processing occur. As shown in Figure 3.2, the Axil 311 workstation is not suitable as a server unless the memory factor is reduced to 0.2 or 0.3 depending on the simultaneous reduction in OS functionality processing factor. Given the rate of improvement in memory, this is very unlikely. Upgrading the CPU to a 88 SpecInt'92 workstation will improve the performance. However, the memory factor cannot be higher than 0.4 to fulfil the server requirements. For powerful workstations with SPECint'92 values at 200 and 400, a memory factor of roughly 0.6 is necessary to fill up a 155 Mbit/s ATM with 65 simultaneous TCP streams.

To sum up, the growth in CPU power is not sufficient to turn a general purpose workstation into a network server that can keep a moderate number of streams going simultaneously. Memory system enhancements are necessary. The effectiveness of the operating system is as illustrated not a major contributing factor.

3.2 Multiple streams – single copy implementation

As illustrated in the previous section, the limited memory speed is a problem for high performance end-systems. A lot of work has been done to come up with solutions to this problem. Most of the proposed solutions are to avoid unnecessary data copy operations in the protocol stack. In one particular solution, which

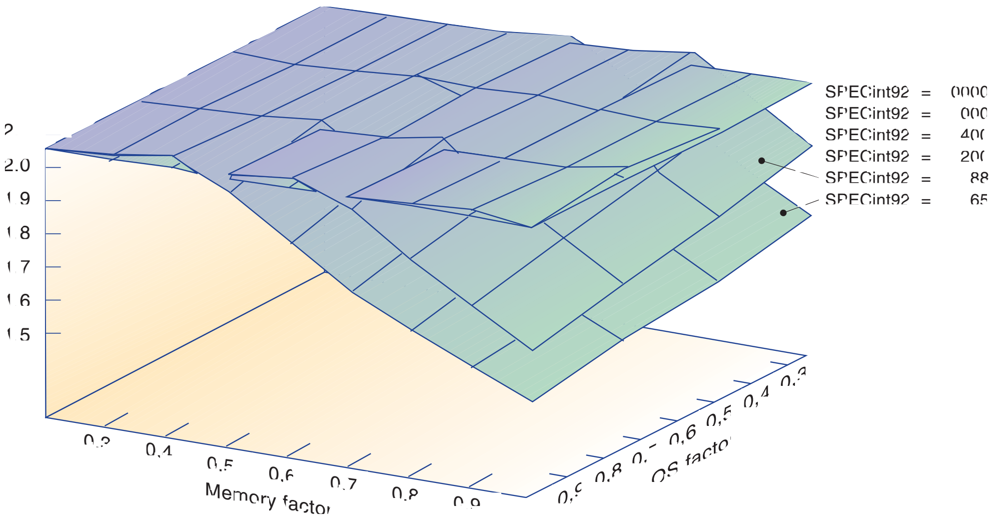


Figure 3.3 Simulations of a single copy implementation

we denote single-copy implementation, the user data is not copied but mapped between user space and system space. The rest of the copy and data touching operations [4] in the protocol stack are still used. Protocol implementations utilizing single-copy techniques will improve the overall end-to-end throughput performance of a given end-system. To predict the influence of single-copy implementations, we modified the simulator. We modified the timing of the uiomove operation by setting the per-byte cost of the uiomove operations in both the sender and the receiver to 0. The per-packet cost was the same, even though the mapping operation in itself will require more processing.

Figure 3.3 shows the average throughput for a single-copy implementation for all six SPECint'92 values. Workstations with SPECint'92 values above 200 will manage to provide an average throughput higher than 2 Mbps.

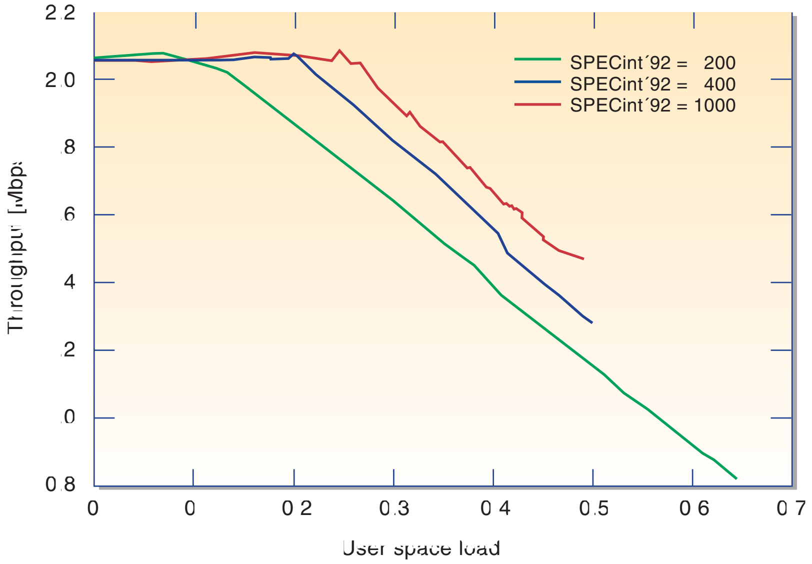


Figure 3.4 Throughput drop as function of additional processing

3.3 Multiple streams – application processing

The simulations we have presented in the two previous sections are modeled after the TCP benchmark ttcp. This benchmark is writing data from the same buffer over and over again during the entire data transfer. Thus, the necessary processing in user space is minimal. In the best case, the user buffer might even be cached during the entire data transfer and thereby improving the throughput significantly.

A server will have to do a lot more processing between each write() system call. Examples are http processing and file retrieval. The results are therefore not valid for a network server. In order to evaluate the influence of user level processing, we modified the simulator to do some artificial processing between each write() system call. The purpose of this modification of the simulator is to demonstrate the effect additional user processing can have on the overall network server throughput.

Figure 3.4 illustrates the average stream throughput as a function of the user space load for three different server configurations. User space load means the amount of processing the network server does in user space during the distribution of 65 TCP streams. All these server configurations have the same memory and operating system parameters, i.e. 0.5 memory factor and 0.7 OS factor. The difference between these systems is the SPECint'92 values which are 200, 400 and 1000 respectively.

As illustrated in Figure 3.4, the user space processing must be lower than 30 % even for a 1000 SPECint'92 TCP server. With user space loads above the knee, maximum achievable throughput for 65 concurrent TCP streams drops linearly.

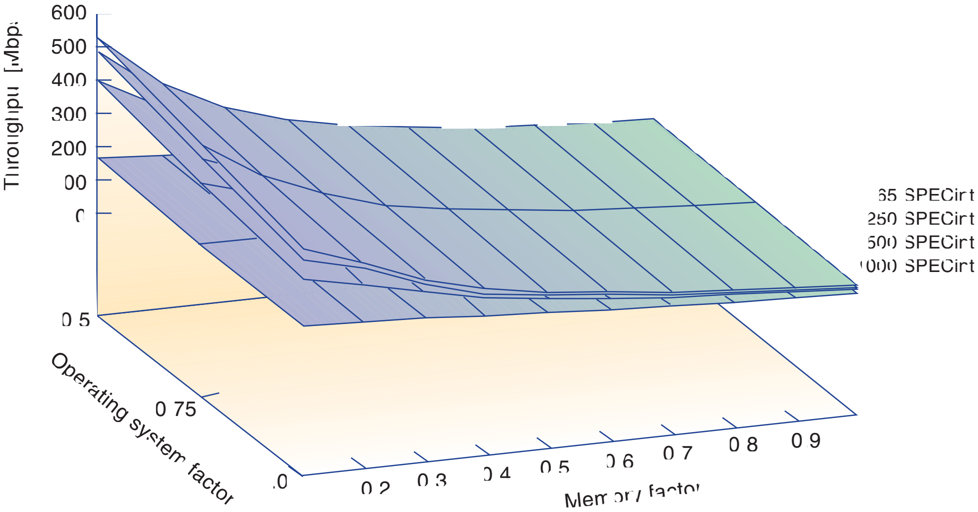


Figure 3.5 622 Mbps ATM – single connection

3.4 Multiple streams – 622 Mpbs

The measurements in the three previous sections have simulated performance over 155 Mbps ATM links. Today, some vendors are delivering 622 Mbps ATM switch and network adapter solutions. We modified the simulator to use the 622 Mbps technology instead. The simulations illustrated in Figure 3.5 show the achievable throughput for a single connection over a 622 Mbps ATM link. As

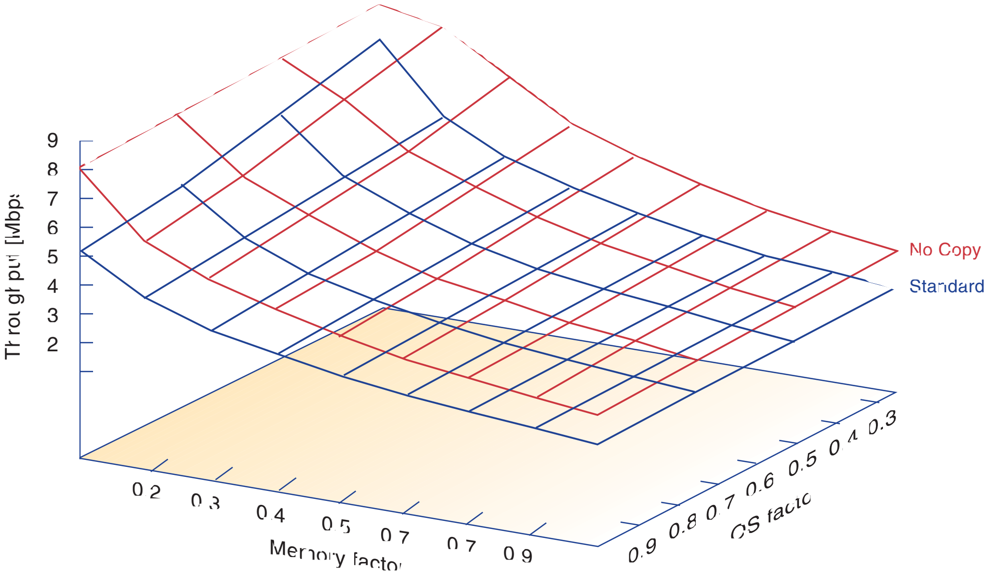


Figure 3.6 Simulations of a 10000 SPECint92 server with 65 simultaneous streams

illustrated in the figure, a significant increase in memory and processor speed is required before the ATM link can be fully utilized by a single connection.

The simulations in Figure 3.6 illustrate the difference between the Standard and NoCopy implementations while distributing data over 65 concurrent connections. The network link is 622 Mbps. The NoCopy performance gives a implementation edge, but still additional increases in both the memory and OS factor are required in order to be able to fill the network link (65 * 8 Mbps).

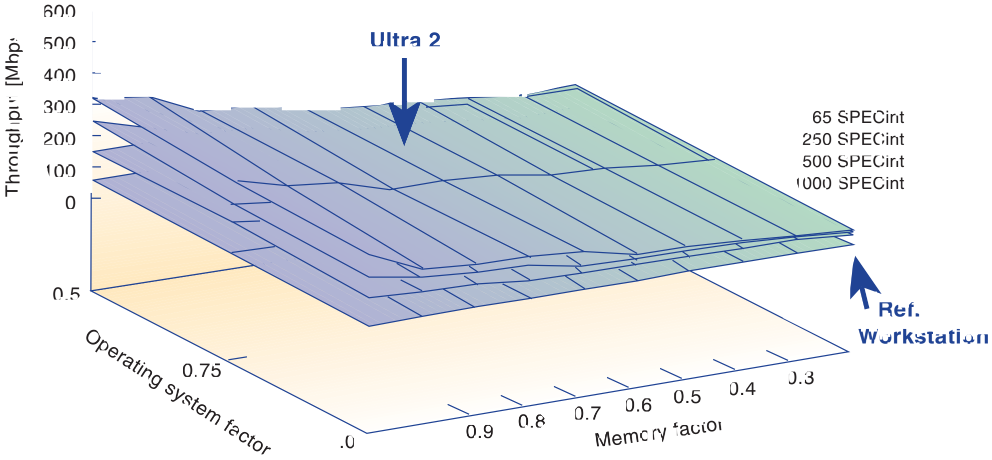


Figure 3.7 Full duplex Gigabit Ethernet throughput – single connection with standard MTU = 1500 byte

3.5 Multiple TCP stream and Gigabit Ethernet

One of the network solutions which will be introduced in local area networks the next few years is Gigabit Ethernet. This technology which will be standardized in 1997 is an extension of the popular Ethernet and Fast Ethernet standards.

Gigabit Ethernet is backward compatible with the existing Ethernet solutions and provides a simple upgrade path for Ethernet users to Gigabit networking. Gigabit Ethernet will offer two different access methods. The most important one is the half duplex CSMA/CD technology used in the other Ethernet technologies. However, Gigabit Ethernet will also offer a full duplex collision-free access method which primarily will be used for interconnecting high end servers, switches and workstations.

One of the problems Gigabit Ethernet will have with the performance in the high bitrate domain is the limited MTU imposed by existing Ethernet solutions. The simulator was modified to use the full duplex access method for the Gigabit Ethernet simulations. The simulation results shown in Figures 3.7 and 3.8 illustrate the performance problems created by the limited Ethernet MTU size for Gigabit networking. In these figures we have indicated where the reference workstation is located and roughly where the new Ultra 2 machine is located in the performance graph. The Ultra machine has a much more aggressive memory system than the Axil 311. Still, the throughput performance will be lower than the link bandwidth for 622 Mbps and Gigabit links.

With the standard MTU, a powerful 1000 SPECint workstation will only be able to achieve 30 % of the Gigabit link speed even with an aggressive memory system. These simulations were done with a standard TCP implementation.

The advantage of using larger MTU sizes is illustrated in Figure 3.8 which shows that the performance can be doubled if the MTU size is increased to 9180 bytes.

4 Discussion

The overall result is the illustration that CPU power alone is not sufficient to turn general purpose workstations into efficient network servers. By efficient we mean a server that is capable of running a full application load while the underlying network is used at peak bandwidth. A combination of growth in CPU power and increase in memory bandwidth is required.

Three inferences can be drawn. The first and obvious one is that memory architecture and protocol architectures geared towards high memory bandwidth will continue to be an important research topic within the network community. The expected compounding growth in CPU power will not be sufficient.

As a direct consequence, when evaluating network servers, the memory architecture and effective bandwidth are as important characteristics as CPU power. Many factors affect the memory bandwidth. At the bottom there is the raw speed of the bus or interconnect. This bandwidth is reduced by overhead of setting up memory transactions. However, as important is how the bandwidth is used. As examples, in older versions of the IP protocol stack, a user packet could be copied up to 7 times before it was transmitted. For the evaluation of protocol performance, the overall memory bandwidth is the important factor; not the underlying bandwidth of the interconnect.

The overall speed of the operating system function is less of an issue. The exceptions are clearly operating system functions that have a direct bearing on memory bandwidth. The latter is typically a challenge when incorporating QoS functions in operating system and protocol stack. Often this is done by adding new protocol stacks and functions without a thought of the effect this will have on the memory bandwidth. As an example, a simple misalignment of a protocol frame may force a copy operation to realign the buffer; the implication is a degradation of the effective memory bandwidth.

The third inference has to do with network architecture. Using general purpose workstations as servers is appealing given that high-end machines always will be equipped with the latest processors. The growth of workstation performance can directly be harnessed into growth in server performance. However, the bandwidth used in most of the simulations in this article was 155 Mbps. At higher bandwidths, e.g. 622 Mbit/s and 2.4 Gbit/s, it is unlikely in the near future that we will see general purpose servers which are able to drive multiple streams at the full bandwidth. Network architectures based on centrally located servers will therefore have a disadvantage; they will not be able to fully utilize the bandwidth, and they will only be able to scale within a limited range. Given a development trend with rapid increases in CPU power and a substantially slower growth in effective memory bandwidth, distributed schemes based on slower workstations will have advantages. Potentially they can scale by adding new servers as the load increases and they have the potential of better utilizing the underlying bandwidth.

From a technical viewpoint, our approach has shown the strengths and weaknesses of the simulation tool. Simulations are useful when the problem cannot be solved by analytic methods or when predictions of future solutions are to be evaluated. The drawback of a simulator is of course to validate that it is representative for the system under study. A substantial part of the effort therefore has to be directed towards the modeling of the system and the validation. The end result is a powerful tool for predicting the protocol performance as a function of CPU power, memory speed, and operating system speed.

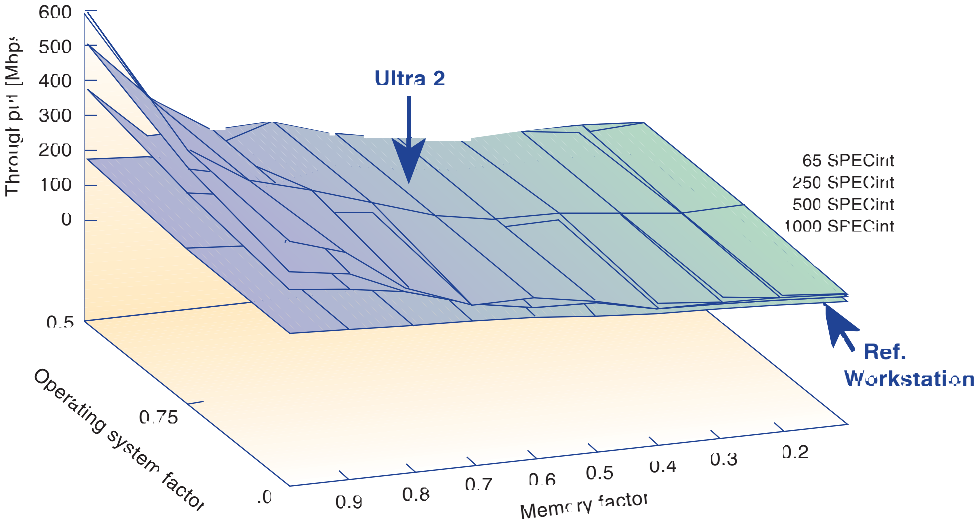


Figure 3.8 Full duplex Gigabit Ethernet throughput – single connection with MTU = 9180 byte

References

1 Amdahl, G M. Validity of the single processor approach to achieving large scale computing capabilities. *Proceedings of the AFIPS 1967 spring joint computer conference,* 30, 483–485, 1967.

2 Moldeklev, K, Klovning, E, Kure, O. TCP/IP behaviour in a high-speed local ATM network environment. *Proceedings of the 19th conference on local computer networks,* Minneapolis, Minnesota, 1994.

3 Pagels, M A, Druschel, P, Peterson, L L. *Cache and TLB effectiveness in processing network I/O.* Department of computer science, University of Arizona, 1994. (TR-94-08.)

4 Kay, J, Pasquale, J. The importance of non-data touching processing overheads in TCP/IP. *Proc. ACM Communications Architectures and Protocols Conf. (SIGCOMM),* San Fransisco, CA, September 1993, 259–269.

5 Ousterhout, J K, Why aren't operating systems getting faster as fast as hardware? *Proceedings of USENIX summer conference,* June 11–15, Anaheim, California, 1990.

6 Anderson, T E et al. The interaction of architecture and operating system design. *Proc. 4th international conference in architectural support for programming languages and operating systems (ASPLOS IV),* April 1991, 108–120. (ISBN 0-89791-380-9.)

7 Catanzaro, B. *Multiprocessor system architectures.* Mountain View, CA, SunSoft Press, 1994. (ISBN 0-13-089137-1.)

8 Lyle, J D. *SBus information, applications and experience.* Berlin, Springer, 1992. (ISBN 0-387-97862-3).

9 Leffler S J et al. *The design and implementation of the 4.3BSD UNIX operating system.* Reading, Mass., Addison-Wesley, 1989. (ISBN 0-201-06196-1.)

10 Clark, D et al. An analysis of TCP processing overhead. *IEEE Communications magazine,* June 1989, 23–29.

11 Anderson, T E et al. A case for NOW(network of workstations). *IEEE Micro,* 15, (1), 54–64, February 1995.

12 Banks, D, Prudence, M. A high-performance network architecture for a PA-RISC workstation. *IEEE JSAC,* 11, (2), 191–202, 1993.

13 Ramakrishnan, K K. Performance considerations in designing network interfaces. *IEEE JSAC,* 11, (2), 203–219, 1993.

14 Wulf, W A, McKee, S A. Hitting the memory wall : implications of the obvious. *Computer Architecture News,* 23, (1), 20–24, March 1995.

15 *FORE Systems Inc. 200-Series ATM adapter : design and architecture,* January 1994.

16 Wright, G R, Stallings, W R. *TCP/IP illustrated volume 2 : the implementation.* Reading, Mass., Addison-Wesley, 1994. (ISBN 0-201-63354-X.)

17 Stallings, W R. *TCP/IP illustrated volume 2 : the protocols.* Reading, Mass., Addison-Wesley, 1994. (ISBN 0-201-63346-9.)

18 Nagle, J. Congestion control in TCP/IP internetworks. *Internet Engineering Task Force,* 1994. (RFC 896.)

19 Sun Microsystems Inc. *Writing device drivers, Part No. 800-3851-10, Rev. A,* March 1990.

For a presentation of the authors, see pages 11 and 28.

Multidestination ARQ protocol for radio LANs

TORE J. BERG

Multidestination automatic repeat request (ARQ) protocols provide reliable transmission of one packet to two or more destinations. The design of a multidestination ARQ protocol for a wireless local area network (LAN) must reflect the operational requirements and environmental constraints to which it is subject. The design of a radio LAN is often based on a decentralized architecture where a random access protocol provides fair access to the radio channel. Here packet collisions cannot be avoided. The probability of having a collision is a function of the network traffic and the use of an ARQ protocol gives rise to a very strong feedback between the traffic and the radio channel quality because lost packets are retransmitted. A degraded radio channel is therefore most likely to be caused by a high level of multiuser interference and one main design challenge is to regulate the use of the radio channel. We propose a multidestination ARQ protocol for a single channel radio LAN that provides a high utilization of the radio channel. An analytical throughput model of the protocol is presented and its performance is illustrated by presenting a case study.

1 Introduction

To satisfy the needs of wireless data networking different methods have been developed for asynchronous accessing a common radio channel. Random access techniques are robust because they do not rely on any special control mechanisms and are easily implemented in distributed systems. Their main disadvantage is the difficulty to find a distributed control mechanism that regulates the user activity such that all links remain above the signal-to-noise ratio (SNR) threshold needed for having successful packet transmissions. The efficiency of these techniques vary with user traffic and radio connectivity. The intention of this paper is to propose an automatic repeat request (ARQ) protocol which provides efficient point-to-multipoint packet switched services in a carrier sense multiple access (CSMA) radio LAN.

Traditionally, ARQ protocols have been divided into three basic types:

a. Stop-and-wait (SW)
b. Go-back-N (GBn)
c. Selective-repeat (SRn).

The SW protocol is usually the least efficient because the channel stays idle between the transmission of the packet and the reception of the acknowledgement from the receiver. However, in a radio LAN other nodes can be given access to the communication channel during this idle period.

[1] to [9] study a series of multireceiver ARQ protocols. One main idea is to send multiple copies of a packet instead of a single copy and some implement an adaptive scheme that, based on the packet loss experienced of previous transmissions, dynamically adjusts the number of copies to send to the receivers. Such a procedure will not give increased performance in a radio LAN using random access.

Many studies assume error free return channels for acknowledgements but for real systems this can only be implemented in exceptional cases. The capability to serve acknowledgement packets is of equal importance as serving data packets because most communications demand reliable delivery of data. This study assumes no error free return channel. Communicating nodes over broadcast channel where acknowledgement and data packet share this channel meet different challenges. One of these challenges is that the ARQ protocol affects the packet loss probability. For example, if the scheduling of data packets are prefixed by a priority delay [10] [11], a system where acknowledgements never experience collisions can be achieved. The first study implementing this is [11]. Here, slotted ALOHA, non-persistent CSMA and 1-persistent CSMA were analysed for point-to-point communications.

This study assumes a fully connected net and uses the network model in Figure 1 for simplifying the analytical treatment. A number of mobile stations, say *n*, receive packets from the users according to some distribution. The packet arrival and the packet length distributions are assumed to be identical for all users. The incoming traffic pattern is uniformly distributed and the path loss is identical for all radio links. The benefits of these choices are that the offered traffic will not be the cause of any non-homogeneous node behaviour and ease the modelling of the network operation.

This paper is organized as follows: The system under consideration is specified in Chapter 2 by employing the layering concept commonly used by radio LAN standards [17] [18]. Firstly, the radio services required to perform packet switching is regarded and then a simple unslotted random access MAC protocol is specified. Finally, Chapter 2 specifies the new logical link control (LLC) protocol providing reliable exchange of point-to-multipoint packets over the radio channel. After completing the specification of the system, the throughput performance is analysed in Chapter 3. A heavy-load model is used, that is, each network node has always one or more packets under service.

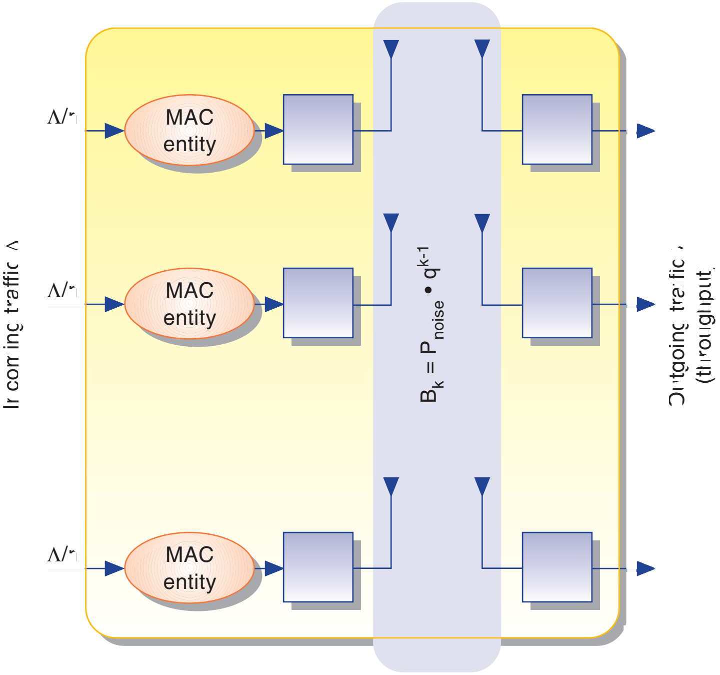

Incoming traffic Λ
Λ/n
Λ/n
Λ/n
MAC entity
MAC entity
MAC entity
$B_k = P_{noise} \bullet q^{k-1}$
Outgoing traffic (throughput)

Figure 1 A model of the radio LAN. The incoming traffic and the radio environment are configured to give identical operating conditions for each node

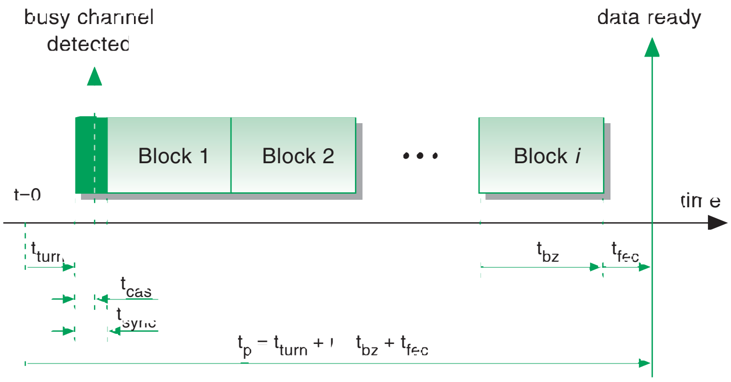


Figure 2 Time diagram for the transmission of a packet on the radio link. A radio switches to transmit mode at $t = 0$ and t_v time units later its neighbours detect a busy channel

The next step is to present throughput results which is the subject of Chapter 4. This chapter also validates the throughput model by presenting simulated results. We close with Chapter 5 which summarizes the design concept in the context of the work on radio LANs and allows us to draw certain conclusions about the main accomplishment and their limitations.

2 The system

The system is specified as a three layer hierarchy following the layering concept commonly in use by radio LAN standards [17] [18]. The layers are the physical layer, the Medium Access Control (MAC) layer and the Logical Link Control (LLC) layer.

2.1 The physical layer

This section specifies the radio services required by the upper layer protocols. First of all, a CSMA protocol requires a “carrier sense” service from the radio which can be implemented by measuring the channel energy level or by using particular codes. Seen from the network level, the “carrier sense” service must be appreciated according to: 1) how fast it can track channel state transitions, and 2) the probability of signalling an incorrect channel state. We continue by considering Figure 2 which illustrates the time diagram for a single packet transmission on the channel. A radio has an idle-to-transmit switching time delay; from the point in time a node decides to transmit to the time instance the signal actually emits from the antenna, is greater than zero. Then the radio wave propagates through the air and reaches the receiver antenna after a delay determined by the distance and the speed of light. Let the sum of these delays be t_{turn}. Generally, the first part of a transmission is a dedicated bit pattern called a preamble with the length t_{sync}. The preamble is needed to synchronize the receiver with the transmitter. A carrier sense time, t_{cas}, is the time necessary to achieve acceptable level of correct determination of the radio channel state; busy or idle. The first opportunity for a receiving radio to detect a transmission is after the time delay $t_{turn} + t_{cas}$. Within this period of time the network receivers are unable to detect a transmission and this time period, defined as the **vulnerable period** t_v, has great impact on the network performance, an issue to be illustrated later.

Imperfect carrier sensing causes two unwanted effects. First of all, an overlapping transmission occurs if the radio misjudges the channel state and signals idle when it is busy, denoted by the probability $p_{cas,I}$, while the network level prepares for a new transmission. The second effect addresses the cases where the radio signals busy while the channel is idle, denoted by the probability $p_{cas,B}$, leading to the abortion of an ongoing packet scheduling. However, the packet will be rescheduled for transmission at a later time. A high $p_{cas,I}$ does more harm because it increases the multi-user interference while a high $p_{cas,B}$ does not[1], but only introduces an additional packet service delay.

Practical use of radio demands implementation of a Forward Error Correction (FEC) coder to reduce the channel bit error rate to a level suitable for data transmissions. Thus, a time delay t_{fec} passes from the last bit received to the time where the packet is available at the network level. We assume use of block oriented FEC such that the user packets have to be divided into an integral number of blocks. If a packet is not of correct length to fulfil this, dummy data must be filled in at the originating side and removed by the destination. When the whole frame is received, an error detection process is applied to the frame and corrupted frames are never delivered to the network level for further processing. For simplicity of presentation, the packet length t_p is defined to also include the radio switching delays and the FEC delay.

Generally, a system is affected by background noise as well as multi-user interference. The latter is normally much higher due to the random access protocol used and depends on a protocol parameter to be described in the sequel. Let B_k be the probability that the first packet transmission succeeds given k–1 overlapping packets. An assumption often used in the literature is $B_{k+1} / B_k = constant = q$ which gives the simple capture model

$$B_k = p_{noise} q^{k-1} \tag{1}$$

[12] presents a scheme to find q for a regular grid network. [13] to [15] look at capture models for other spatial distributions. One major network design challenge is to control the multi-user interference represented by k. This becomes evident by regarding the term q^{k-1} in (1). A real system has $q < 1$ and then the packet success drops fast with increasing k. A system where the first transmission sustains any number of overlapping transmissions is said to have **perfect capture** ($q = 1$) as opposed to the **zero capture** case ($q = 0$), where any overlapping transmission leads to demodulation failure. Radios providing **non-perfect capture** fulfil the relationship $0 < q < 1$. By using spread spectrum signalling, high q-values can be achieved.

2.2 The MAC layer

To date numerous CSMA protocols have been presented in the literature. Their difference stems from the use of different random access delay distributions when accessing the channel, behaviour under retransmission and some are slotted while others are unslotted. To have a reasonable complete treatment of the many CSMA protocols we refer to [19]. We use a simple unslotted CSMA protocol with uniformly distributed scheduling delay D based on two parameters

[1] *At least not directly but possibly indirectly due to increased channel traffic.*

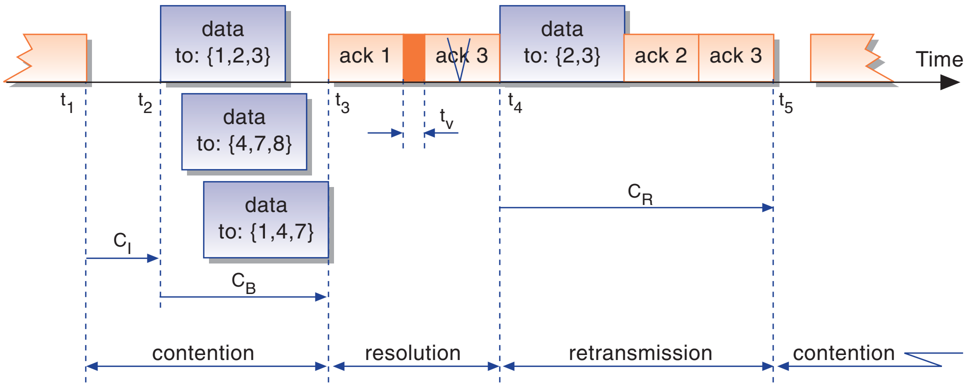


Figure 3 Example of a delivery cycle. The first data packet with the destination set {1,2,3} (d_{max} = 3) is successfully received by node 1 and 3. Acknowledgement 3 is lost

$$D(p_f, t_u) = p_f \cdot t_v + UniformDistribution[0, t_u] \qquad (2)$$

The first is the priority delay access factor (p_f) and the second is the upper bound (t_u) for the uniform distribution. p_f is a natural number and t_u is a real number greater than zero. Then D has the cumulative density function (cdf) $F_D(t)$ given by

$$F_D(t) = \left[\begin{array}{ll} 0 & \text{for } t < p_f \cdot t_v \\ \frac{1}{t_u}(t - p_f \cdot t_v) & \text{for } t \in [p_f \cdot t_v, t_u + p_f \cdot t_v] \\ 1 & \text{for } t > t_u + p_f \cdot t_v \end{array} \right. \qquad (3)$$

2.3 The LLC layer

The LLC protocol shall provide reliable multipoint addressing, that is, each data packet is transmitted on the broadcast channel until all the addressed nodes have returned an acknowledgement to the originator. The destination addresses of the user data packets may change randomly subject to the restriction that the number of destinations must be kept fixed to a size d_{max}. Network throughput improvement is achieved in two ways. Firstly, by utilizing the MAC protocols priority service, collision-free transmissions of acknowledgement packets are achieved. Secondly, by giving precedence to data packets that have been successfully delivered to at least one destination, but for which some remain, packet collisions can be eliminated in the retransmission(s) to follow.

We will now give an overview of the link layer. Appendix 1 gives the detailed specification of the link layer protocol. The description is assisted by defining three channel states as seen by an outside observer: the contention state, the resolution state and the retransmission state. Consider the time instance where the system is in the contention state at the end of a transmission, time t_1 in Figure 3. The nodes having data packets ready for service draw a random access delay $D(d_{max}, t_u)$ defined by (2) (A6)[2]. The node getting the shortest delay starts to transmit at t_2. The duration $t_2 - t_1$ is referred to as the channel idle period C_I. Within the vulnerable period $[t_2, t_2 + t_v]$ additional nodes may start to transmit because they have not yet detected a busy radio channel. The last transmission ends at t_3 and completes the channel busy period C_B.

At time instance t_3 all nodes behave as described at t_1 except the nodes that successfully received the packet and are members of its destination set (A8). They construct an acknowledgement to be scheduled for transmission at $D(idx - 1{,}0)$, where idx is the node index within the destination set (A10). This gives the acknowledgement packets a precedence ordering after node location in the destination set of the data packet and all acknowledgements are given precedence over data packets. The first acknowledgement gets zero access delay while the last takes the delay $(d_{max} - 1) \cdot t_v$.

The transition from contention to resolution state occurs when a winning node can be declared, that is, when at least one of the addressed nodes successfully returns an acknowledgement (A16). Whenever the source node receives an acknowledgement in this state, the data packet is immediately scheduled for retransmission with a priority delay that takes into account the number of outstanding acknowledgements (A57, A60). For each of the missing acknowledgements a time gap of size t_v occurs.

The winning node must keep track of the correct point in time to retransmit while the nodes having acknowledgements pending, must transmit in the correct order despite the action of their predecessors. The difficulty arises when the background noise destroys an acknowledgement so that its “from”-field is useless (A41). This problem is solved by introducing a timer with a fixed duration t_v. At the timer expiration time (A36), the node shall decrement its priority if no channel activity is detected. Only nodes that have a pending acknowledgement shall activate this timer. The winning node needs only to decrement its priority each time it receives a packet with check sum error (A58).

The transition from resolution to retransmission state occurs the first time the winning node retransmits the data packet. The retransmission period of length C_R constitutes a random number of retransmissions and acknowledgements, determined by the background noise. In Figure 3 delivery is completed at t_5 (A50) after only one retransmission in the retransmission state.

[2] *(Ax) refers to line number x in Appendix 1.*

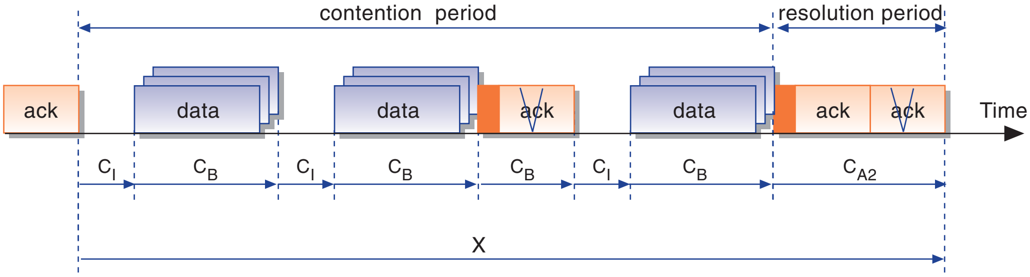


Figure 4 Illustration of contention and resolution periods

3 The throughput model

This chapter considers the throughput performance of the multi-destination ARQ protocol specified. The restrictions introduced above provide for identical node behaviour. For the tractability of the mathematics we analyse the network under a heavy-load assumption. In this state every network node always have a packet to serve and then the packet arrival distribution at the radio channel level is determined by the MAC random access distribution alone. Beside simplifying the mathematics, the heavy-load state is the most important state to analyse because random access systems are known to suffer from instability at increasing load. The analysis is started by a few additional definitions, see Figure 4. We define:

C_{A1} $\overset{\triangle}{=}$ the time delay between the end of a busy period and the end point of the last acknowledgement in the contention period. If no acknowledgement occurs, then $C_{A1} = 0$.

C_{A2} $\overset{\triangle}{=}$ the duration of the resolution period.

N_A $\overset{\triangle}{=}$ the number of acknowledgements successfully returned after a busy period.

N_D $\overset{\triangle}{=}$ the number of data packets successfully delivered at the end of a busy period.

σ_A $\overset{\triangle}{=}$ the probability that at least one acknowledgement is successfully returned

$= P(N_A > 0)$.

X $\overset{\triangle}{=}$ the total duration of the contention period and the resolution period.

The mean time passed until all the destinations have received the data packet is $E[X + C_R]$ where C_R is the retransmission period length. Each packet delivered contains N_P blocks and d_{max} such packets are delivered. Hence, the number of blocks delivered per time unit is

$$d_{max}E[N_P] \,/\, E[X + C_R] \qquad (4)$$

We normalize (4) by dividing by d_{max} and the physical layer transfer rate $1/t_{bz}$ and get

$$\lambda \overset{\triangle}{=} \text{normalized network throughput, } 0 \leq \lambda \leq 1$$
$$= t_{bz}E[N_P] \,/\, E[X + C_R]. \qquad (5)$$

From Figure 4 it is easy to realize that

$$X = \sum_{i=1}^{K_A - 1} \{C_I + C_B(|\text{all ack. lost}) + C_{A1}\} \qquad (6)$$
$$+ C_I + C_B(|\text{at least one ack succeeds}) + C_{A2}$$

when K_A is the number of contention periods passed until at least one acknowledgement is successfully returned. These periods are independent and the K_A is geometrically distributed with the mean $1/\sigma_A$. Therefore, the first moment of X can be written as

$$E[X] = \left(\frac{1}{\sigma} - 1\right)(E[C_I] + E[C_B|N_A = 0] + E[C_{A1}])$$
$$+ E[C_I] + E[C_B|N_A > 0] + E[C_{A2}] \qquad (7)$$
$$= (E[C_I] + E[C_B])/\sigma_A + \left(\frac{1}{\sigma_A} - 1\right)E[C_{A1}] + E[C_{A2}]$$

Here $E[C_B|N_A = 0] > E[C_B|N_A > 0]$ because when failures occur the mean number of transmitting nodes is higher. The idle period C_I starts at the beginning of the delivery cycle and ends with the beginning of the first transmission. Under the heavy-load condition n nodes independently draw their random access delay, the smallest of these is the first node to transmit and therefore

$$F_{C_I}(t) = P\left(\min\left[\underbrace{D, ..., D}_{n}\right]\right) = 1 - [1 - F_D(t)]^n \qquad (8)$$

The average channel idle period is then simply

$$E[C_I] = \int_0^\infty [1 - F_{C_I}(t)]\, dt$$
$$= \int_0^\infty [1 - F_D(t)]^n\, dt \qquad (9)$$
$$= \frac{t_u}{n+1} + p_f \cdot t_v$$

To find the cdf of the busy period is more complicated. However, the conditional first moment of C_B is found in [12] as

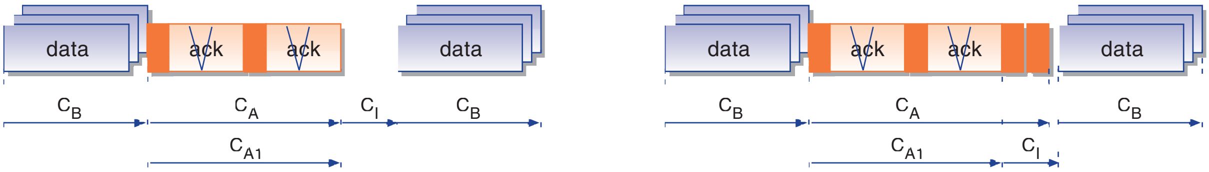


Figure 5 Illustrating the acknowledgement phase in the contention state that may end with an acknowledgement (the left figure), or with a number of time gaps N_{t_v} each of size t_v

$$
\begin{aligned}
&E[C_B|K_{Tx}=k]\\
&=t_{turn}+t_{bz}\sum_{r=0}^{a}\left(1-\left(\frac{r}{a}\right)^k\right)\\
&+t_v\sum_{r=1}^{a}\left(\frac{r}{a}\right)^k\frac{k-r+r(1-1/r)^k}{k}
\end{aligned}
\tag{10}
$$

when N_P is discrete uniformly distributed such that $P(N_p = r) = 1/a$. Unconditioning gives

$$E[C_B]=\sum_{k=1}^{n}E[C_B|K_{Tx}=k]P(K_{Tx}=k) \tag{11}$$

where K_{Tx} is the number of simultaneous transmissions in the contention period.

For each addressed node that does not receive the data packet, a time gap of size t_v occurs, or t_{ack} if the data packet is correctly received but the acknowledgement is destroyed by background noise. Let C_A be this length, see Figure 5.

Conditioning on that $N_D = d$,$(d_{max} – d)$ destinations do not receive the packet but d destinations do, we have

$$E[C_A|N_D = d] = d_{max}t_v + (t_{ack} – t_v)d \tag{12}$$

leading to

$$E[C_A] = d_{max}t_v + (t_{ack} – t_v)E[N_D] \tag{13}$$

Each node immediately draws a random scheduling delay when the radio channel becomes idle and therefore C_{A1} is shorter than C_A, see Figure 5. Let N_{t_v} denote the number of t_v-slots after the last acknowledgement in the contention phase. Then

$$
\begin{aligned}
&E[C_{A1}] = E[C_A \mid N_A = 0] – t_vE[N_{t_v} \mid N_A = 0]\\
&\text{and } E[C_{A2}] = E[C_A \mid N_A > 0]
\end{aligned}
\tag{14}
$$

where

$$
\begin{aligned}
&E[N_{t_v} \mid N_A = 0] =\\
&\sum_{d=0}^{d_{\max}}E[N_{t_v}|N_A=0,N_D=d]P(N_D=d|N_A=0)=\\
&\frac{1}{1-\sigma_A}\sum_{d=0}^{d_{\max}}E[N_{t_v}|N_D=d]P(N_A=0|N_D=d)P(N_D=d)
\end{aligned}
\tag{15}
$$

Inserting (14) into (7) gives

$$
\begin{aligned}
&E[X] = (E[C_I] + E[C_B] + E[C_A]) / \sigma_A\\
&– t_v(1 / \sigma_A – 1)E[N_{t_v} \mid N_A = 0]
\end{aligned}
\tag{16}
$$

Given $N_D = d$, d acknowledgments and $(d_{max} – d)$ t_v-time gaps will follow and then it is easy to realize that

$$
\begin{aligned}
&P(N_{t_v} = a \mid N_D = d)\\
&=\left[\begin{array}{ll}
1 & \text{for } d=0, a=d_{\max}\\
\frac{d}{d_{\max}-a}\prod_{x=0}^{a-1}\left(1-\frac{d}{d_{\max}-x}\right) & \text{for } a\in[0,d_{\max}-d], d>0\\
0 & \text{otherwise}
\end{array}\right.
\end{aligned}
\tag{17}
$$

N_A is zero in the contention period. When N_A becomes one or greater, at least one acknowledgement is successfully returned and the resolution period started at the end of the last busy period. When d destinations receive the data packet, all return an acknowledgement each succeeding with the probability p_{noise}. Therefore,

$$P(N_A=a|N_D=d)=\binom{d}{a}p_{noise}^{a}(1-p_{noise})^{d-a} \tag{18}$$

leading to

$$\sigma_A=1-\sum_{d=0}^{d_{\max}}(1-p_{noise})^dP(N_D=d) \tag{19}$$

Now we develop the pdf for N_D which, by the total probability formula, can be written as

$$
\begin{aligned}
P(N_D=j)&=\sum_{k=1}^{n}P(N_D=j|K_{Tx}=k)P(K_{Tx}=k)\\
&=\sum_{k=1}^{n}\sum_{m=0}^{d_{\max}}P(N_D=j|K_{Tx}=k,D_{Tx}=m)\\
&\cdot P(D_{Tx}=m|K_{Tx}=k)P(K_{Tx}=k)
\end{aligned}
\tag{20}
$$

where

K_{Tx} $\overset{\triangle}{=}$ the number of simultaneous transmissions in the contention period

D_{Tx} $\overset{\triangle}{=}$ the number of nodes in the destination set that also transmit.

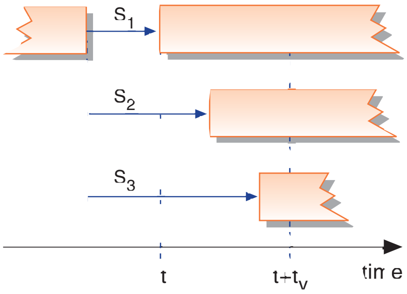


Figure 6 The first node transmits at t. At t + t_v all the non-transmitting nodes detect a busy channel

Given that m of the addressed nodes are transmitting, the number of idle destinations that may successfully receive the data packet is ($d_{max} – m$). A transmitting node can, of course, never receive a packet. The probability of successful demodulation for each independent node is given by the capture probability B_k (1) and by the binomial distribution we conclude (for $m \in [0, k – 1]$ and $m \leq d_{max}$) that

$$P(N_D = j \mid K_{Tx} = k, D_{Tx} = m) =$$

$$\left(\begin{array}{ll} 1 & j = d_{\max}, m = 0 \\ \binom{d_{\max}-m}{j} B_k^j (1 - B_k)^{d_{\max}-m-j} & 0 \leq j < d_{\max} - m \\ 0 & \text{otherwise} \end{array} \right. \quad (21)$$

If we look at the first transmission addressed for d_{max} nodes under the condition $K_{Tx} = k$ and $D_{Tx} = m$, there are ($n – k$) idle nodes, ($d_{max} – m$) of the addressed nodes are idle and ($k – 1$) other nodes are also transmitting. Each destination is randomly drawn from a population of size ($n – 1$) and the source node selects d_{max} of these (sampling without replacement). Then

$$P(D_{Tx} = m \mid K_{Tx} = k) =$$

$$\left(\begin{array}{ll} \binom{k-1}{m} \binom{n-k}{d_{\max}-m} / \binom{n-1}{d_{\max}} & \text{for } 0 \leq m \leq k-1, m \leq d_{\max} \\ 0 & \text{otherwise} \end{array} \right. \quad (22)$$

which is a hypergeometric distribution. (21) and (22) give

$E[N_D \mid K_{Tx} = k, D_{Tx} = m] = (d_{max} – m)B_k$ and

$$E[D_{Tx} | K_{Tx} = k] = d_{\max} \frac{k-1}{n-1}$$

leading to

$$E[N_D] = d_{\max} \sum_{k=1}^{n} \frac{n-k}{n-1} \cdot B_k \cdot P(K_{Tx} = k) \quad (23)$$

Let $S_1 \leq S_2 \leq ... \leq S_n$ be the order statistics obtained by ordering the node scheduling intervals (random access delays). The node getting the smallest scheduling interval starts to transmit after the delay S_1, see Figure 6.

If $S_1 = t$ and $S_k \in [t, t + t_v]$ then at least k simultaneous transmissions have occurred. The event $K_{Tx} = k$ occurs when one of the n nodes schedules a transmission at t, ($k – 1$) nodes schedule a transmission within $[t, t + t_v]$ and the remaining ($n – 1) – (k – 1$) nodes wait till after $t + t_v$. All nodes behave independently up to $t + t_v$ and therefore

$$P(K_{Tx} = k) = \binom{n-1}{k-1} \int_0^{\infty} [F_D(t + t_v) - F_D(t)]^{k-1}$$

$$[1 - F_D(t + t_v)]^{n-k} n f_D(t) dt \quad (24)$$

where $F_D(t)$ is the cdf (3) and by inserting that expression we have

$$P(K_{Tx} = k) = n \binom{n-1}{k-1} \int_0^{d_{\max} t_v} 0 dt$$

$$+ n \binom{n-1}{k-1} \int_{d_{\max} t_v}^{t_u + (d_{\max}-1) t_v} \left(\frac{t_v}{t_u} \right)^{k-1} [1 - (t - (d_{\max} - 1) t_v) / t_u]^{n-k}$$

$$/ t_u dt + n \binom{n-1}{k-1} \int_{t_u + (d_{\max}-1) t_v}^{t_u + d_{\max} t_v} [1 - (t - d_{\max} t_v) / t_u]^{k-1} 0^{n-k} / t_u dt$$

$$= \binom{n}{k-1} \left(\frac{t_v}{t_u} \right)^{k-1} \left(1 - \frac{t_v}{t_u} \right)^{n-k+1} + \left(\frac{t_v}{t_u} \right)^n 0^{n-k} \quad (25)$$

At this point only one stochastic variable remains, namely the retransmission period C_R which is the subject of the next section.

3.1 The retransmission period

The multidestination ARQ protocol switches to the retransmission state at the end of the resolution period if one or more acknowledgements are missing. This section considers the

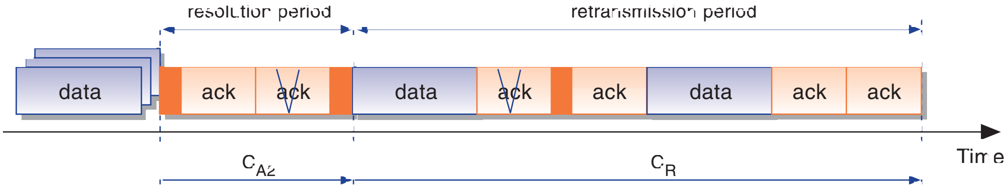


Figure 7 The resolution and retransmission periods as seen by an outside observer

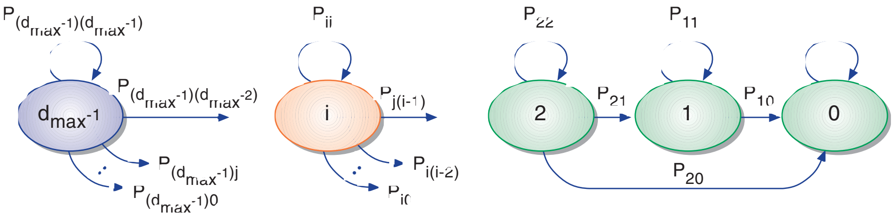


Figure 8 The number of missing acknowledgements form a simple Markov chain with one absorbing state

length of the retransmission period. First we need to define the following two stochastic variables:

$N_A^{(0)}$ $\overset{\triangle}{=}$ the number of missing acknowledgements at the end of the resolution period

$N_A^{(i}$ $\overset{\triangle}{=}$ the number of missing acknowledgements at the end of retransmission number i, $i \geq 1$

and then we can write

$$P\left(N_A^{(1)} = j | N_A^{(0)} = i\right) = \binom{i}{j} p_{noise}^{2(i-j)} \left(1 - p_{noise}^2\right)^j \qquad (26)$$

because each addressed node independently returns a successful acknowledgement only if both the data packet and the acknowledgement packet arrive free from bit errors (no co-channel interference occurs in the retransmission period).

If $N_A^{(i)} = 0$, the retransmission period ends after i retransmissions. If $N_A^{(0)} = 0$, all destinations will have received the first packet transmitted in the last contention period.

For simplicity of description, we define

$\overline{C_R}^{(i)} \overset{\triangle}{=} E[C_R \mid N_A^{(0)} = i]$

and by total probability

$$E[C_R] = \sum_{i=1}^{d_{\max}-1} \overline{C_R}^{(i)} \cdot P\left(N_A^{(0)} = i\right) \qquad (27)$$

The relation between $N_A^{(0)}$ and N_A is simply $N_A^{(0)} = d_{max} - N_A(| \, N_A > 0)$ and the pdf for $N_A^{(0)}$ is

$$\begin{aligned} P(N_A^{(0)} = i) &= P(N_A = d_{max} - i \mid N_A > 0) \\ &= P(N_A = d_{max} - i) \,/\, \sigma_A \qquad \text{for } 0 \leq i < d_{max} \end{aligned} \qquad (28)$$

We have a simple discrete time Markov chain with the transition probabilities P_{ij} given by (26). The chain has the transient states $\{1, ..., d_{max} - 1\}$ and the absorbing state $\{0\}$, see Figure 8. First step analysis yields

$$\overline{C_R}^{(i)} = \sum_{j=0}^{i} P_{ij} \cdot \overline{C_R}^{(i \to j)} + \sum_{j=1}^{i} P_{ij} \cdot \overline{C_R}^{(j)} \qquad (29)$$

where $\overline{C_R}^{(i \to j)}$ is the average transition time delay from state i to state j. This is a set of equation of the form

$$\begin{bmatrix} 1 - p_{11} & 0 & 0 & 0 & 0 \\ -P_{21} & 1 - p_{22} & 0 & 0 & 0 \\ -P_{31} & -P_{21} & 1 - P_{22} & 0 & 0 \\ ... & ... & ... & ... & 0 \\ -P_{m1} & -P_{(m-1)2} & ... & ... & 1 - P_{mm} \end{bmatrix} \cdot \begin{bmatrix} \overline{C_R}^{(1)} \\ \overline{C_R}^{(2)} \\ ... \\ \overline{C_R}^{(m)} \end{bmatrix} = \begin{bmatrix} g(1) \\ g(2) \\ \\ g(m) \end{bmatrix} \qquad (30)$$

where

$$m \overset{\triangle}{=} d_{\max} - 1 \text{ and } g(i) \overset{\triangle}{=} \sum_{j=0}^{i} P_{ij} \cdot \overline{C_R}^{(i \to j)}$$

and is easily solved by Gaussian elimination when an expression for $\overline{C_R}^{(i \to j)}$ has been found. A retransmission will always contain one data packet of the length $t_{turn} + t_{bz} \cdot E[N_P]$. If a destination fails to demodulate the data packet, $\overline{C_R}^{(i \to j)}$ is increased by t_v. Otherwise, $\overline{C_R}^{(i \to j)}$ will be increased by t_{ack}. Given that j failures occur, the average duration introduced by the two delay components are $j \cdot t_{ack} p_{noise}(1 - p_{noise}) \,/\, (1 - p^2_{noise})$ and $j \cdot t_v(1 - p_{noise}) \,/\, (1 - p^2_{noise})$. This leads to

$$\overline{C_R}^{(i \to j)} = t_{turn} + t_{bz} \cdot E[N_p] + (i - j)t_{ack} + j \cdot \frac{t_v + p_{noise} \cdot t_{ack}}{1 + p_{noise}} \qquad (31)$$

because $(i - j)$ is the number of successful addresses. They will always introduce a t_{ack} time delay.

This completes the deduction of the throughput model and the next chapter gives numerical examples and validates the results against simulation results.

4 Case study

The objectives of this chapter are to present numerical throughput results and validate the theoretical throughput model against simulated results. We start by defining the fixed parameters for the case study, see Table 1. [12] argues that $q = 0.82$ is a realistic approximation for a spread spectrum receiver with processing gain 11 dB operating in a complete network topology.

Figure 9 Throughput as a function of the random access delay distribution parameter t_u. Simulated results are presented as vertical arrows and represent 90 % confidence intervals.
The input parameters are:
Fixed packet length $N_p = 1$
Network size $n = 9$ (as dotted curves) and $n = 25$ (as solid lines)
The destination set size values are {2,3,5,8,15,20,24} ($d_{max} < n$)

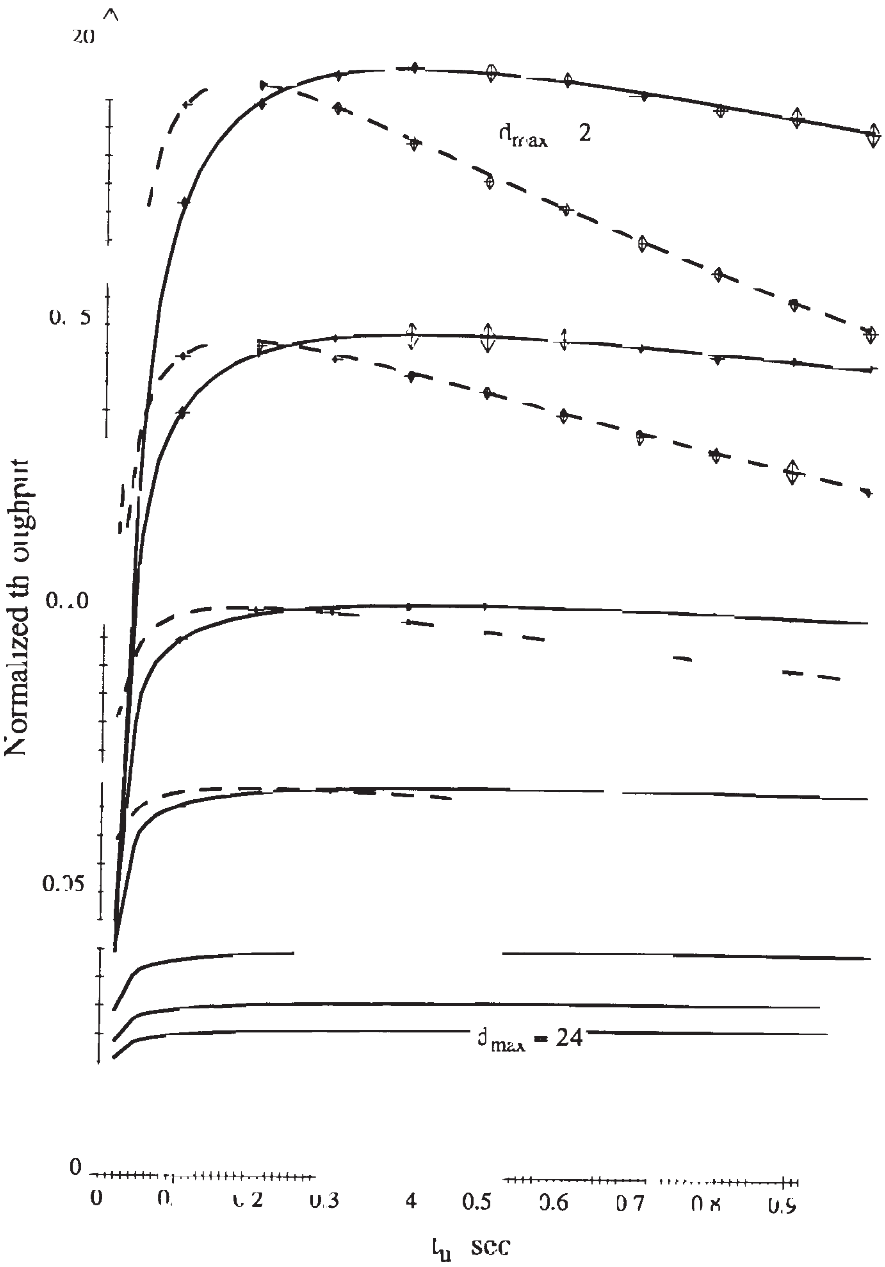


Table 1 Default radio parameters

parameter	value
"carrier sense" delay (t_{cas})	6.4 msec
block length (t_{bz})	48 msec
synchronization time (t_{sync})	6.4 msec
radio turn time (t_{turn})	3.6 msec
FEC processing delay (t_{fec})	0
CAS failing probability ($p_{cas,I}$, $p_{cas,B}$)	(0,0)
q	0.82
p_{noise}	0.9

Table 2 Variables and their range

variable name	values
network size, n [number of nodes]	9, 16, 25
destination set size, d_{max} restricted by $d_{max} \leq n - 1$	{2,3,5,8,15,20,24}
t_u [sec]	0.1 to 1.0 in steps of 0.1
Length of user data packets, N_P [number of blocks]	Fixed: $N_P = 1$, $N_P = 4$ or $N_P = 7$ Uniformly distributed: $N_P \in \{1, 2, 3, 4, 5, 6, 7\}$

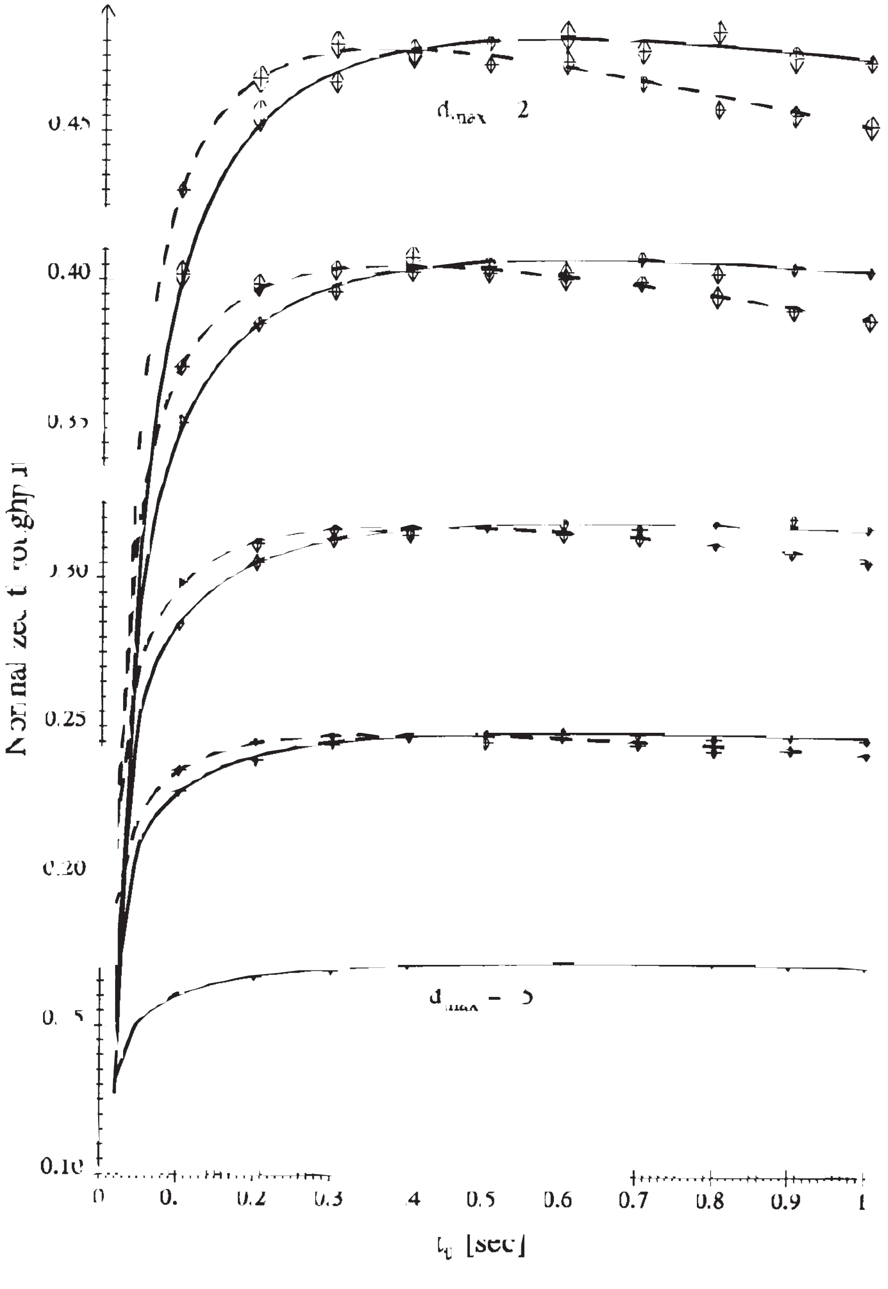


Figure 9 (continues)
The input parameters are:
Fixed packet length $N_p = 7$
Network size n = 9 (as dotted curves) and n = 16 (as solid lines)
The destination set size values are {2,3,5,8,15}
($d_{max} < n$)

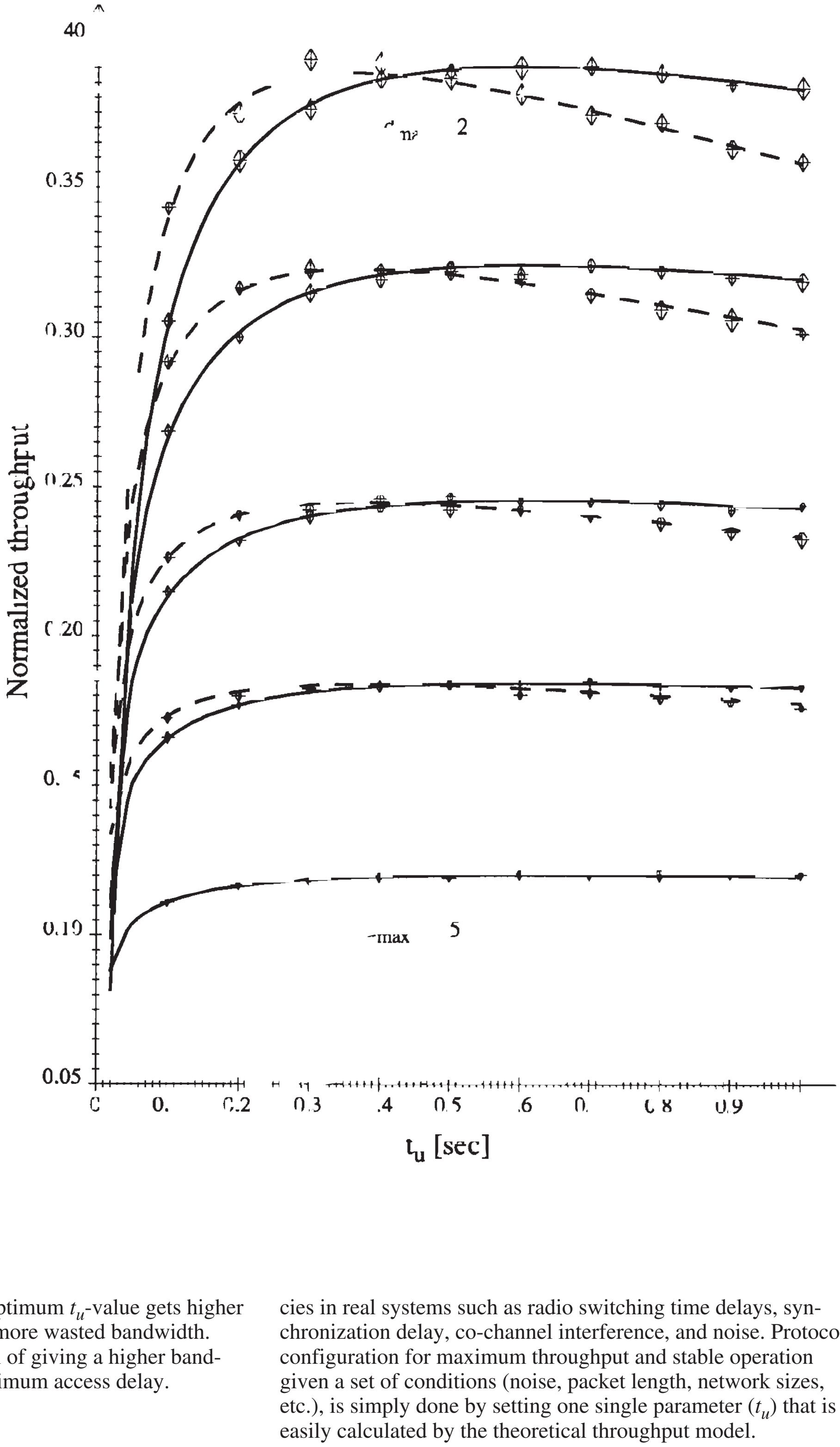
Table 2 shows the system variable values used. A wide range of collision rates is covered. The highest rate occurs at ($n = 25$, $t_u = 0.1$) where $E[K_{Tx}] = 3.5$ while the lowest collision rate happens at ($n = 9$, $t_u = 1.0$) where $E[K_{Tx}] = 1.09$.

Figure 9 presents a selected set of the throughput results as generated from the input values defined in Table 2. Simulated results are plotted as 90 % confidence intervals, and the theoretical and simulated results show excellent conformity.

The MAC protocol parameter t_u has a significant impact on the network performance. As t_u decreases, the probability of having two or more simultaneous transmissions increases, while the average channel idle time period becomes shorter. With increased t_u the opposite results. At a given user traffic and a set of system parameters, a t_u-value exists that gives an optimum balance between the channel idle time and the channel capacity wasted due to collisions. Increasing d_{max} leads to increased probability of addressing other transmitting nodes and therefore, to maintain maximum throughput, t_u must also be increased.

The normalized throughput decreases with increasing d_{max} because we used d_{max} as a normalizing factor in (5). However, the utilization of the radio channel transmission capacity becomes better with increasing d_{max} due to the fact that one single transmission can be delivered to more than one destination while the overhead remains fixed.

Figure 9 (continues)
The input parameters are:
Random packet length $N_p \in \{1, ..., 7\}$
Network size $n = 9$ (as dotted curves) and $n = 16$ (as solid lines)
The destination set size values are $\{2,3,5,8,15\}$
($d_{max} < n$)



As the packet length increases, the optimum t_u-value gets higher because each packet loss represents more wasted bandwidth. Yet longer packets have the potential of giving a higher bandwidth utilization by selecting the optimum access delay.

5 Conclusion

A multidestination ARQ protocol for single channel single hop radio LANs has been proposed. The decentralized protocol utilizes the radio channel capacity effectively for the operating environment described. The protocol is able to handle deficiencies in real systems such as radio switching time delays, synchronization delay, co-channel interference, and noise. Protocol configuration for maximum throughput and stable operation given a set of conditions (noise, packet length, network sizes, etc.), is simply done by setting one single parameter (t_u) that is easily calculated by the theoretical throughput model.

The design of radio networks is complicated by the presence of hidden nodes. A pair of stations are hidden if they are not within the radio range of each other. In networks where hidden nodes exist, the carrier sense function in the originating node fails to determine the state of the receiving side when the destination node is captured on a packet from a hidden node. If a network contains many links suffering from the hidden node problem, a reservation protocol may give better performance than a random access method [16]. Beside precluding the CSMA-operation, the presence of hidden nodes implies that the network topology does not constitute a complete graph and packet relaying must be provided.

The user data packets are assumed to include randomly changing destination sets containing exactly d_{max} elements. However, the protocol can handle randomly changing destination set sizes less than or equal to d_{max}, but this complicates the analysis and is subject for further study.

References

1 Ram Chandran, S, Lin, S. Selective-repeat-ARQ schemes for broadcast links. *IEEE Trans. Commun.,* 40, (1), 1992.

2 Hayashida, Y. Throughput analysis of tandem-type go-back-N ARQ scheme for satellite communications. *IEEE Trans. Commun.,* 41, (10), 1993.

3 Wang, J L, Silvester, J A. Optimal adaptive multireceiver ARQ protocols. *IEEE Trans. Commun.,* 41, (12), 1993.

4 Gopal, I, Rom, R. Multicasting to multiple groups over broadcast channels. *IEEE Trans. Commun.,* 42, (7), 1992.

5 Towsley, D, Mithal, S. A selective repeat ARQ protocol for a point to multipoint channel. *INFOCOM 87 proceedings,* 1987.

6 Sabnani, K, Schwartz, M. Multidestination protocols for satellite broadcast channels. *IEEE Trans. Commun.,* COM-33, (3), 1985.

7 Towsley, D. An analysis of a point-to-multipoint channel using a go-back-N error control protocol. *IEEE Trans. Commun.,* COM-33, (3), 1985.

8 Gopal, I S, Jaffe, J M. Point-to-multipoint communication over broadcast links. *IEEE Trans. Commun.,* COM-32, (9), 1984.

9 Metzner, J J. An improved broadcast retransmission protocol. *IEEE Trans. Commun.,* COM-32, (6), 1984.

10 Onunga, J O, Donaldson, R W. Performance analysis of CSMA with priority acknowledgements (CSMA/PA) on noisy data networks with finite user population. *IEEE Trans. Commun.,* 39, (7), 1991.

11 Tobagi, F A, Kleinrock, L. The effect of acknowledgement traffic on the capacity of packet-switched radio channels. *IEEE Trans. Commun.,* COM-26, (6), 1978.

12 Berg, T J, Emstad, P J. Performance of packet switched services in spread spectrum radio networks. *Telektronikk,* 91, (4), 1995, 140–152. (ISSN 0085-7130.)

13 Soroushnejad, M, Geraniotis, E. Probability of capture and rejection of primary multiple-access interference in spread-spectrum networks. *IEEE Trans. Commun.,* 39, (6), 1991.

14 Lau, C T, Cyril Leung, C. Capture models for mobile packet radio networks. *IEEE Trans. Commun.,* 40, (5), 1992.

15 Wong, V, Leung, C. Capture probability in a mobile packet radio system. *IEEE Trans. Commun.,* 40, (10), 1992.

16 Bharghavan, V et al. MACAW : a media access protocol for wireless LAN's. *Proceedings of ACM SIGCOMM,* 1994.

17 ETSI. *High Performance Radio Local Area Network (HIPERLAN).* European Telecommunications Standard Institute, 1994.

18 IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *Draft Standard IEEE 802.11,* P802.11/D5, IEEE Standards Department, 1996.

19 Rom, R, Sidi, M. *Multiple Access Protocols : Performance and Analysis.* Springer, 1990. (ISBN 0-387-97253-6.)

Tore J. Berg was, at the time of writing this article, Research Scientist at Telenor R&D, Kjeller. He has since left the company for other employment.

Appendix 1

This appendix gives the detailed specification of the link protocol as implemented in each network node.

```
Protocol( y: Event )
   case NodeState is
      when contentionPeriod =>
         case y is
            when DT_PDU to send =>
               AckSet = DT_PDU.DestinationSet
               schedule DT_PDU at $d_{max}*t_v$ + Uniform[0,$t_u$]
            when incoming DT_PDU =>
               if addressed to me then
                  AckSet = DT_PDU.DestinationSet
                  priority = index( D, i ) - 1
                  state = waitToIdle
                  if idle channel then Protocol( chTransitionBusyIdle ) fi
               fi
            when incoming ACK_PDU =>
               if addressed to me then
                  state = resolutionPeriod
                  Protocol( y )
               fi
         end case
      when waitToIdle =>
         case y is
            when chTransitionBusyIdle =>
               state = outgoingAckPending
               if priority > 0 then startTimer($t_v$) fi
               schedule ACK_PDU at $t_v$*priority
         end case
      when outgoingAckPending =>
         case y is
            when incoming ACK_PDU =>
               priority = index( AckSet - {...,ACK_PDU.FROM}, i ) - 1
               if priority > 0 then startTimer($t_v$) fi
               schedule ACK_PDU at $t_v$*priority
            when the node transmits the acknowledgement =>
               stopTimer()
               state = contentionPeriod
            when timeout =>
               if carrier detected then return fi
               priority = priority - 1
               if priority > 0 then startTimer($t_v$) fi
               schedule ACK_PDU at $t_v$*priority
            when check sum error =>
               priority = priority - 1
               if priority > 0 then startTimer($t_v$) fi
               schedule ACK_PDU at $t_v$*priority
         end case
      when resolutionReriod =>
         case y is
            when incoming ACK_PDU =>
               AckSet = AckSet - {ACK_PDU.From}
               if AckSet = {} then
                  delete DT_PDU
                  state = contentionPeriod
                  return
               fi
               priority = |AckSet - {...,ACK_PDU.From}|
               DT_PDU.DestinationSet = AckSet
               schedule DT_PDU at $t_v$*priority
         when check sum error =>
               priority = priority - 1
               schedule DT_PDU at $t_v$*priority
         end case
   end case NodeState
end Protocol
```

Status

International research and
standardization activities
in telecommunication

Editor: Per Hjalmar Lehne

Per Hjalmar Lehne is Research Scientist at Telenor Research & Development, Kjeller

email address: per.lehne@fou.telenor.no

Introduction

PER HJALMAR LEHNE

The *International Telecommunications Union (ITU)* still plays an important role in global standardisation in the telecommunications area. The ITU divides the work into four year Study Periods, of which the period 1993–1996 has recently ended. It has been decided that ITU-T shall take all responsibility of the standardisation work, also the radio aspects, while ITU-R shall concentrate on regulatory matters.

In this issue of the Status section we look at the results from the recently ended Study Period. We have four contributions addressing the work on the Standardisation Sector, ITU-T. All of the papers address the changes at the start of the new Study Period, 1997–2000.

In the first paper, Mr. Arve Meisingset presents the results and decisions from the second *World Telecommunications Standardisation Conference – WTSC-96* which was held in Geneva in October 1996. Some key figures of the ITU-T are given. In order to deal with the rapid changes in the telecommunications area, the Study Group structure of the ITU-T has been reorganised. The paper addresses this as well as changes in the management structure and work procedures.

Multimedia is one of the present ‘buzzwords’ in IT and telecommunications. In the second paper, Mr. Trond Ulseth presents how the topic has been studied in the Study Period 1993–1996. A new Study Group, SG16 – *Multimedia Services and Systems*, is established. Previously, the work on multimedia was performed in several study groups. The goal of this reorganisation is that the work on this topic, which is experiencing a growing interest, shall be more efficient and better co-ordinated. The paper addresses important results from the work and an overview of the relevant recommendations is given.

The last two papers are both written by Mr. Meisingset. The first deals with the results and activities of ITU-T Study Group 10 of which Mr. Meisingset is vice chairman. SG10 deals with *Languages and general software aspects for telecommunication systems*. SG10 had its first meeting in the new Study Period in April–May 1997. The Study Group is responsible for technical languages, the methods of their usage and other issues related to the software aspects of telecommunication systems. The paper contains a comprehensive list of the main goals for SG10 and all Questions. Mr. Meisingset is also acting rapporteur of Question 3/10 on *Software platforms and middlewares for the telecom domain*.

In the last paper, Mr. Meisingset presents the ITU-T Recommendation Z.360 on *Graphic GDMO*[1]. GDMO deals with the TMN[2]-part of the OSI specifications, and the graphic GDMO provides an overview for TMN specifiers, implementors and users. The managed object classes are presented together with a realistic example of the usage of GDMO.

Table 1 List of previous contributions to the Status section

Issue No.	Title of published papers	Authors
4.93	Service definitions	Ingvill H. Foss
4.93	Radio communications	Ole Dag Svebak
4.93	Transmission and switching	Bernt Haram
4.93	Intelligent Networks	Endre Skolt
4.93	Broadband	Inge Svinnset
1.94	Terminal equipment and user aspects	Trond Ulseth
1.94	Signal processing	Gisle Bjøntegård
1.94	Telecommunications Management Network	Ståle Wolland
1.94	Teletraffic and dimensioning	Harald Pettersen
1.94	Data networks and open system communications	Berit Svendsen Mette Røhne
2.94	The TINA Consortium	Tom Handegård
2.94	Telecommunications languages and methods	Arve Meisingset
2.94	Message Handling Systems	Geir Thorud
2.94	Security	Sverre Walseth
3.94	EU's research programme ACTS	Eliot J. Jensen
3.94	UPT- Service concept, standardisation and the Norwegian pilot service	Frank Bruarøy Kjell Randsted
3.94	Status report form ITU-TSB, SG 1	Elisabeth Tangvik
3.94	Future mobile communications	Ole D. Svebak
3.94	The CIO project	Gabriela Grolms
4.94	Eurescom work on ATM network studies and the European ATM pilot network	Inge Svinnset
4.94	Terminal equipment and user aspects	Trond Ulseth
4.94	Telecommunications Management Network	Ståle Wolland
2/3.95	Documents types that are prepared by ETSI	Trond Ulseth
2/3.95	ATM traffic activities in some RACE projects	Harald Pettersen
2/3.95	The Public Network Operator Cipher project	Øyvind Eilertsen
4.95	Migration from today's IN, TMN and B-ISDN towards TINA	Thomas Rødseth
4.95	Methods for dimensioning of Intelligent Networks	Terje Jensen
4.95	Mobility applications integration in Intelligent Networks	Geir Olav Lauritzen
4.95	EURESCOM P408 “Pan-European TMN – experiments and field trial support”	Pål Kristiansen
1.96	Global Information Infrastructure	Arve Meisingset
1.96	FPLMTS – IMT2000: Communication anywhere, anytime	Dag Fredrik Bjørnland
1.96	The Inmarsat Mobility Management System	Arvid B. Johannessen Arild Flystveit
3/4.96	COST 219: Future telecommunications and teleinformatics facilities for disabled people and elderly	Per Helmersen
3/4.96	COST 235: Radiowave propagation effects on next generation fixed-services telecommunication systems	Agne Nordbotten
3/4.96	COST 231: Evolution of land mobile radio (including personal) communications	Per Hjalmar Lehne Rune Harald Rækken

[1] *GDMO: Guidelines for the Definition of Managed Objects.*

[2] *TMN: Telecommunications Management Network.*

International Telecommunication Union
World Telecommunication Standardization Conference – WTSC-96

ARVE MEISINGSET

Introduction and overview

WTSCs are arranged between each four year Study Period of the ITU-T:

- The first WTSC was arranged in Helsinki in 1993
- The second WTSC was held in Geneva 9–18 October 1996
- The third WTSC is planned in Canada, October 2000.

WTSC is the highest body of the standardisation sector, ITU-T, of ITU. Read more on the ITU organisation and work in http://www.itu.ch/.

WTSC-96 was attended by 530 representatives from 80 member states. 40 Recognised Operating Agencies, including Telenor AS, and 26 Scientific and Industrial Organizations attended independent of their national delegation. The Japanese bench impressed with about 50 representatives from tens of industrial and scientific organisations.

In accordance with a decision in the ITU council, administrations represent Member States, while Recognised Operating Agencies and Scientific and Industrial Organizations are called Sector Members.

Member States (MS)	186
Sector Members (SM)	263
Increase of SM in 1995	36
Recognised Operating Agencies (RoAs)	112
Scientific and Industrial Organizations (SiOs)	151
Applied for SG participation	60 %
Participation in SG meetings	30 %

Figure 1 Participation in ITU-T

	ITU	ITU-T
Member States	88 %	45 %
Sector Members	12 %	55 %

Figure 2 Financing of ITU

	Number of Recommendations
Approved by Resolution no. 1	407
Approved at WTSC-96	32

Figure 3 ITU-T production figures

WTSC-96 was smoothly run in fewer days than WTSC-93, much due to good preparation by TSAG. Read more about the role of TSAG below. The results from WTSC-96 will be published in a separate ITU-T Book.

Figure 1 provides some data on the size of ITU-T.

Some Study Groups, such as SG11 Signalling requirements and protocols, draw a very large number of participants, while other SGs are declining and experience strong competition from regional and world wide research organisations, standardisation bodies, consortia and fora.

Figure 2 provides information on the financing of ITU.

Currently, members are providing financial contributions to ITU on a voluntary basis. This is not likely to change soon. However, some sector members want to earmark a part of their contribution to a specific sector within ITU. This has not been agreed on by the members.

Recommendations have been approved by Member States by a reply to Circular letters from TSB, Telecommunication Standardization Bureau, or by direct approval at WTSC. Figure 3 provides an overview of the use of the two alternative approval procedures.

The approved number of Recommendations is similar to that of the previous Study Period; however, the number of pages produced is drastically increasing. Draft Recommendation G.691 was not approved, due to patent claims by one of the members. Draft Recommendation J.110 was returned to the Study Group.

New Study Group Structure

WTSC-96 decided to dissolve SG1, Service Definition, and SG14, Modems and transmission techniques for data, telegraph and telematic services. USA proposed to dissolve SG10, Languages and general software aspects for telecommunication systems, as well. However, this proposal was opposed by several members.

A large set of Questions from SG1 was transferred to the enlarged SG2, Network and Service Operation. Questions from SG7, SG11 and SG15 were transferred to a strengthened SG4, TMN and Network Maintenance. WTSC-96 established a new SG16, Multimedia services and systems. This organisational change indicates that ITU is able to adapt to new problems and markets, and is not stuck in traditional telecommunication technologies.

Figure 4 provides an overview of the new structure of ITU-T, together with an indication of chairmen, vice chairmen and Telenor co-ordinators for each Study Group. Several SGs were assigned a co-ordinating role for a specific area. This role is called a Lead Study Group.

As seen from Figure 4, most chairmen and vice chairmen come from industrial countries, which make up the 30 % of the members who participate actively in the SGs' work.

ITU has been accused of being a rather slow and bureaucratic organisation, and has felt the competition of more flexible con-

Group	Name	Chairman	Vice chairmen	Telenor co-ordinator
TSAG	Telecommunication Standardization Advisory Group	G Fishman, USA	J Fanjul, E I Kreinguel, Rus K S Park, Kor R F Brett, Can N Kisrawi, Syr	Arve Meisingset, R&D tech. co. Bjørn Sandnes, KU strategy co.
JRG GII	Joint Rapp. Group on Global Information Infrastr.	B W Moor, G		
SG2	Network and service operation.	G Gosztony, Hng J Martorey, F R Blane, G	A Lewis, Can	Arne Østlie, Nett
SG3	Tariff and accounting principles including related ...	T Matsudaira, J A M al Tiwaniy, Oma	W Lucas, G	Per Wien, Global Services
SG4	TMN and network maintenance	D Sidor, USA A Rojdestvensky, Rus	N Fujii, J	Knut Johannesen, Nett
SG5	Protection against electromagnetic environment effects	G Meineri, I	G Varju, Hng	Kjell Gulliksen, Nett
SG6	Outside plant	L Molleda, E		
SG7	Data networks and open systems communications	H Bertine, USA Y H Lee, Kor V Ossipov, Rus	Y Hramatsu, J	Bror Mathisen, Nett
SG8	Telematics	W Staudinger, D A Macchioni, I	A Pugh, G	
SG9	Television and sound transmission	J L Teijerina, E H Murakami, J	R Green, USA	Carl K Nordahl, Nett
SG10	Languages and general software aspects ...	A Sarma, D	A Meisingset, Nor	Arve Meisingset, R&D
SG11	Signalling requirements and protocols	S Kano, J W Vandenbroeck, Bel Ph Distler, F	E A Matarazzo, B	Tormod Egeland, Nett
SG12	End-to-end transmission performance of netw. and ...	M Cao, Chn J Y Monfort, F	C Dvorak, Usa	Jon Rugelbak, R&D
SG13	General network aspects	B W Moore, G K Asatani, J F Lucas, F	J Luechtford, Can	Bjørn Netland, Nett
SG15	Transport networks, systems and equipment	P Wery, Can G Bonaventura, I	M Yamashita, J	Sverre Myren, Nett
SG16	Multimedia	P A Probst, Sui F Tosco, I G Helder, USA	J Magill, I	Helge Sandgrind, R&D

Figure 4 New ITU-T structure

sortia and fora. However, this view was to some degree opposed by statements from the Internet Engineering Task Force in the Global Information Infrastructure Seminar in Geneva, 1996: Development and approval of Internet ‘standards’ take some years, as well. Adding to this picture, fora and consortia wish to use ITU-T for rubber stamping, to give their results world-wide authority.

To meet the competition, WTSC-96 approved the principle of establishing Focus Groups. Focus Group may have participants who are not ITU members and develop answers in a short time period, using their own way of working. However, if the results are to be approved as Recommendations, the same approval procedure as for other Recommendations must be applied.

Management structure

TSAG, Telecommunication Standardization Advisory Group, was established in WTSC-93, and proved very useful for preparation of WTSC-96. This conference delegated authority to TSAG to

- develop A-series Recommendations to be approved by Member States
- consider and initiate the use of co-ordination groups
- establish temporary groups, such as Focus groups.

TSAG acts between WTSCs. The intention is to make TSAG become a permanent body by decisions in the 1998 Plenipotentiary Conference.

Resolution 16 from the Kyoto Plenipotentiary conference asks the directors of the standardisation bureau, TSB, and the radio bureau, RB, to study the distribution of work between their two sectors. The European Telecommunication Network Operators, ETNO, are pushing the ITU to move all standardisation work out of the Radio Sector, ITU-R, and concentrate all standardisation work within the Standardisation Sector, ITU-T. Other regions are opposing this transfer.

Work procedures

The old approval procedure of Recommendations, Resolution No. 1, has been split into a core Resolution No. 1, Rules of Procedure of the ITU Telecommunication Standardization Sector, and a new Recommendation A.1, Work methods for Study Groups of the ITU Telecommunication Standardization Sector (ITU-T). Resolutions can be approved by WTSC every four years, while Recommendations can be approved by Member States at any time.

According to the new Resolution No. 1, a draft Recommendation must first be unopposed in a Study Group plenary, then the Member States are consulted by a Circular letter before final approval of the Recommendation in a following Study Group plenary meeting. Only Member States can vote in these decisions; however, Sector Members dominate the Study Group meetings and have the right to speak. (There is already agreement in TSAG to give Sector Members the right to vote on ‘technical’ Recommendations in Study Group meetings. However, this future procedure is neither worked out nor yet approved. There is currently no agreement on a new future approval procedure for Recommendations of the ‘regulatory’ type.)

Previously, the Circular letter to Member States was posted after the last Study Group meeting. According to the new procedure, Member State opposition can be changed in the final Study Group meeting, and the total approval period is reduced by 4–5 months. Also, the translation to all ITU languages may be postponed until after the final approval.

New Questions can be approved by the Study Group, if consensus by both Member States and Sector Members is achieved, and minimum four members will provide active support to the work. Hereby, Sector Members have already got voting power within the ITU.

Technical results

ITU-T is by the G7 countries assigned responsibility to undertake world wide standardisation of the Global Information Infrastructure (GII). This is carried out through the Joint Coordination Group for GII, where also ISO and IEC participate. Study Group 13 is Lead Study Group for GII. The European Telecommunications Network Operators (ETNO) have contributed to the identification of the following priority areas for GII:

- Terminal and personal mobility
- Accessibility ‘anytime’, ‘anywhere’
- Diversity of services
- Privacy/data security
- Definition of an integrated global framework reference model
- Middleware standardisation
- Network standardisation
- Applications.

Telecommunication operators / administrations in developing countries typically get 60 % of their income from international traffic and use this income to build their national infrastructure, like the industrial countries have been doing in the past. Therefore, service operators providing call-back services are a great threat to the financing of new infrastructure in the developing countries. Hence, many developing countries have banned call-back services by national law and want ITU support for this. However, the industrialised countries have introduced international competition regimes which are in conflict with the current interests of the developing countries. New technologies, like low orbit satellites and handheld satellite terminals can represent a further threat.

References

1 Lillebø, A L. *ITU : World Telecommunication Standardization Conference : WTSC-96.* Telenor Konsernstab utredning, 063.861, 08.11.96.

For a presentation of the author, see page 93.

ITU-T multimedia standardisation activities

TROND ULSETH

Organisation of work in the 1993–1996 study period

In the 1993–1996 study period questions addressing multimedia topics have been studied by the following ITU-T Study Groups,

• Study Group 1	Service Definition
• Study Group 8	Terminals for telematic services
• Study Group 12	End-to-end transmission: performance of networks and terminals
• Study Group 13	General Network Aspects
• Study Group 14	Modems and transmission techniques for data, telegraph and telematic services
• Study Group 15	Transmission systems and equipment.

Important results achieved

Study Group 1 has been working on multimedia related services under two questions. Question 20/1 is addressing Audiovisual Multimedia Services. A general framework recommendation (F.700) and a recommendation on videophone service in the PSTN (F.723) have been approved using Resolution No. 1 procedure. Question 21/1 (New services for the Broadband ISDN) has produced ITU-T Recommendation F.732 (Multimedia conference services in the B-ISDN) which was approved by WTSC in October 1996.

Study Group 8 has been working on three multimedia related questions,

• Q 10/8	Audiographic Conferencing
• Q 11/8	Protocols for interactive audio-visual services
• Q 16/8	Common components for Image and Audio Communication.

Question 10/8 is one of the major study questions on multimedia services and protocols. Although the title of Q 10/8 was Audiographic Conferencing, i.e. simultaneous speech and data, the scope of the work has been extended to multimedia conferencing, i.e. simultaneous transmission of speech, video and data/graphics. The results of the work are the T.120-series recommendations. These recommendations are presented in *Telektronikk* 3/4.1996 [1]. The recommendations approved so far are T.120, T.121, T.122, T.123, T.124, T.125, T.126 and T.127.

At WTSC-96 ITU-T recommendation T.171 and T.174, which are the results of the study under Q 11/8, were approved.

The study of Q 16/8 focuses on still image applications. The latest results are modifications to existing recommendations, T.84 and T.85.

A single Study Group 12 question is addressing multimedia, Q22/12 Audiovisual quality in multimedia services. Three recommendations addressing this topic have been approved, P.910, P.920 and P.930.

Study Group 13 has been studying network capabilities for the support of multimedia services in 64 kbit/s ISDN and B-ISDN. It is planned to issue two draft recommendations on general networks and service specific requirements this year (1997).

The work of Study Group 14 relevant for multimedia applications are high speed modem technology, and procedures for simultaneous data and voice communication on the PSTN.

A significant result is the approval of ITU-T Recommendation V.34 standardising of modem operating at data signalling rates up to 28.8 kbit/s. This modem is used for the PSTN multimedia terminal standardised in ITU-T Recommendation H.324.

Other important recommendations recently approved are,

• V.8	on procedures for starting sessions of data transmission
• V.8bis	on procedures for the identification and selection of common modes of operation between DCEs and between DTEs
• V.61	on a simultaneous voice plus data modem also known as V.asvd
• V.70	on procedures for the simultaneous transmission of data and digitally encoded voice signals, also known as V.dsvd-s
• V.75	on DSVD terminal control procedures, also known as V.dsvd-c.

Study Group 15 has been the most important Study Group in the standardisation of multimedia systems and services. The Study group has been working on questions addressing,

- *Speech coding*
  The latest recommendations approved are G.723 and G.729. G.723 specifies a speech coder operating at 5.3 kbit/s and 6 kbit/s. The main application is PSTN multimedia communication standardised in ITU-T Recommendation H.324. G.729 is specifying a speech coder operating at 8 kbit/s.
- *Video coding*
  Recently ITU-T recommendation H.263 is approved. This recommendation specifies a video coding algorithm that can work from 9.6 kbit/s and upwards.
- *Framing and in-band signalling for audiovisual and multimedia communication*
  New recommendations for systems operating on non-ISDN environment, but including B-ISDN, have been approved. These recommendations include H.222.0, H.222.1, H.223 and H.245. The recommendations for ISDN environment have been revised.
- *System aspects of audiovisual and multimedia terminals*
  New recommendations approved are H.321 (Adaptation of H.320 terminals to B-ISDN), H.322/H.323 (Systems and terminals for LANs with/without a guaranteed quality of service) and H.324 (Low bitrate communication).

A new study group is established

As can be seen the multimedia standardisation activities of ITU-T have been carried out by several study groups. A special co-ordination group, JCG on AVMMS (Audiovisual/Multimedia Services) was set up in WTSC-93 to co-ordinate the ITU-T work on multimedia standardisation. The ITU-T Study

Table 1 Organization of ITU-T SG 16

Working Party 1/16 (Low rate systems)	
Chair:	Mr. J. Magill (Lucent Technologies, United Kingdom)
Q4/16	Modems for switched telephone network and telephone type leased circuits
Q5/16	ISDN terminal adapters, and interworking of DTEs on ISDNs with DTEs on other networks
Q6/16	DTE-DCE interchange circuits
Q7/16	DTE-DCE protocols
Q8/16	DCE/DCE protocols
Q9/16	Text telephony
Q10/16	Testing
Q18/16	Interaction of high-speed voiceband data systems with signal processing equipment in the public-switched telephone network.
Working Party 2/16 (Services and high rate systems)	
Chair:	Mr. F. Tosco (CSELT, Italy)
Q1/16	Audiovisual/multimedia services
Q2/16	Interactive Multimedia Information Retrieval Services (MIRS)
Q3/16	Data protocols for multimedia conferencing
Q11/16	Circuit switched network (CSN) multimedia systems and terminals[1]
Q12/16	B-ISDN multimedia systems and terminals
Q13/16	Packet switched multimedia systems and terminals
Q14/16	Common protocols, MCUs and protocols for interworking with H.300-series terminals
Working Party 3/16 (Signal processing)	
Chair:	Mr. Simão F. Campos Neto (COMSAT Labs., USA)
Q15/16	Advanced video coding
Q19/16	Extension to existing ITU-T speech coding standards at bit rates below 16 kbit/s
Q20/16	Audio and wideband coding in public telecommunication networks
Q21/16	Encoding of speech signals at bit rates around 4 kbit/s
Q22/16	Software and hardware tools for signal processing standardisation activities

Groups 1, 2, 3, 7, 8, 9, 11, 12, 13, 14 and 15 (i.e. most of the ITU-T Study Groups) participated. Study Group 15 was the lead study group.

Some people felt that the JCG on AVMMS was not effective to actually co-ordinate the work on multimedia. Several solutions have been considered, and TSAG finally proposed to the World Telecommunication Standardisation Conference held October 1996 (WTSC-96) to establish a new study group on multimedia. WTSC-96 accepted the proposal. The terms of reference of the new Study Group 16 is:

> *Responsible for studies relating to multimedia service definition and multimedia systems, including the associated terminals, modems, protocols and signal processing.*
>
> *Study Group 16 is the Lead Study Group on Multimedia services and systems.*

The chairman of the study Group is Mr. P.-A. Probst, Switzerland.

Two of the questions allocated to this study group, Q16/16 (Harmonization of multimedia systems, applications and services) and Q16/17 (AVMMS co-ordination), will be discussed in the study group plenary meetings.

The establishment of the Working Parties of Study Group 16 (Multimedia services and systems) was proposed as shown in Table 1.

The questions allocated to WP 1/16 have, after the modification made at the first meeting of the Study Group, their origin from Study Group 14. Important questions for this working party are the extension of ITU-T recommendation V.34 (V.34Q) and simultaneous voice and data technology in general.

The majority of the questions allocated to WP 2/16 have their origin from WP 1/15. Two questions, Q 2/16 and Q 3/16 are continuation of work carried out by Study Group 8 in the 1993–96 study period, and Q 1/16 is a continuation of work in Study Group 1.

All the questions allocated to WP 3/16 have their origin from WP 3/15.

One of the reasons for the proposal to establish a new study group on multimedia was to solve the co-ordination problems between Study Group 14 and Study Group 15 on the principles for simultaneous voice and data communication. Study Group 14 has developed V.61 and V.70, and has been working on an extension to ITU-T recommendation V.34. In parallel, Study Group 15 has developed ITU-T Recommendation H.324 on low bitrate multimedia communication which is based on ITU-T Recommendation V.34. Thus, several recommendations are standardising solutions for similar applications. This situation will neither be to the benefit of the users nor to the manufacturers of terminals and systems.

Although the work is carried out in two separate WPs, the co-ordination problem will be simplified compared with the 1993–96 study period, but the need for harmonisation of the recommendations and the applications still exists and will be a challenge for the new study group.

[1] The invitation to the first meeting of SG 16 proposed to allocate this question to WP 1/16. However, the question has a lot of commonalities with questions allocated to WP 2/16. The first plenary meeting of SG 16 decided to allocate question 11/16 to WP 2/16.

A second challenge for Study Group 16 is to develop standards for communication between terminals and systems in an heterogeneous environment where the networks, the control protocols and the signal processing elements may be different. An example making reference to the ITU-T H-series recommendations is given in Figure 1.

For multiparty communication the complexity of the scenarios increases, e.g. conferences where the participants are attached to different networks or the terminals are supporting different bandwidths.

A third challenge is to standardise communications between terminals/systems designed for different types of applications, e.g. conversational, retrieval or broadcasting services.

The variety of both network protocols, in-band protocols and media processing algorithms highlights the need for procedures for negotiation to identify a common mode of operation or at least modes of operation which can be understood by the other participant(s). Work is going on in this direction both for H-series terminals and V-series modems.

Internet is a new area in ITU-T. Most of the Internet standards so far have been developed under the umbrella of IETF. However, audio and video coding algorithms developed by ITU-T (CCITT) have been used in Internet applications for years. A new trend is ITU-T standardised protocols which can be used by terminals and systems for Internet communication. Examples are ITU-T Recommendation H.323 and the T.120 series recommendations for data conferencing. Several ITU-T Study Group 16 study questions are considering Internet and IP issues as well as issues related to traditional public/private telecommunication network.

Apart from Study Group16 multimedia is studied by some other ITU-T Study Groups. The most important are probably Study Group 11 and Study Group 13 on access networks and access network protocols for multimedia. It is also worth mentioning the work of Study Group 9 (Television and sound transmission).

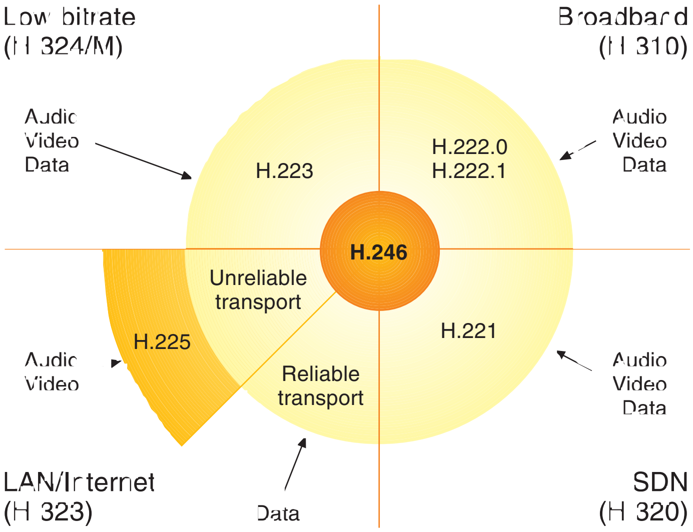


Figure 1 Scope of draft ITU-T Recommendation H.246

Reference

1 Ulseth, T. Protocols for multimedia multiparty conferencing : a presentation of the ITU-T T.120 series recommendations. *Telektronikk,* 92, (3/4), 1996, 95–104.

Trond Ulseth is Project Manager at Telenor R&D, Kjeller.

*email address:
trond.ulseth@fou.telenor.no*

Languages and general software aspects for telecommunication systems

ARVE MEISINGSET

This paper provides a condensed presentation of ITU-T Study Group 10 results and activities after its first Study Group meeting, 28 April – 6 May 1997, of the Study Period 1997 – 2000.

Study Group 10

Chairman:	Amardeo Sarma, Deutsche Telekom
Vice chairman:	Arve Meisingset, Telenor AS
Counsellor:	Audi Ganguli, ITU-T TSB
Secretary:	Isabelle Garde, ITU-T TSB

Responsible for technical languages, the methods for their usage and other issues related to the software aspects of telecommunication systems.

SG10 collaborates with other standardisation organisations, such as ISO and ETSI.

SG10 collaborates with external fora, such as the Object Management Group (OMG), SDL Forum Society and informally with Tina-C.

SG10 provides active support to the ITU-T TSAG Quality standards programme.

SG10's work plan is organised into Questions and GII projects (Global Information Infrastructure).

SG10 will continue work on developing an SG10 strategy. Ericsson proposes the following priorities:

1) Maintenance of existing SG10 languages and harmonisation with other ITU languages
2) Next generation notations for analysis and design, e.g. harmonisation with OMG UML
3) Software architectures and APIs for telecom systems.

Z.110 will be extended to provide a common framework for use of all formal languages within ITU.

Recommendations applicable for the whole Study Group:

Z.110 (1988)	Criteria for the use and applicability of formal description techniques.

SG10 input to the GII Work Plan

SG10 recommends that a Quality programme be established for the GII, on which Q4/10 can provide guidance.

SG10 proposes the following GII projects be established:

1) Middleware for multimedia. Q1/10 is proposed to become Lead Body.
2) Object-oriented environment. Q6/10 is proposed to become Lead Body.
3) Operating environment and user interfaces experienced by the GII end-users or customers. Q2/16 is proposed to become Lead Body.
4) Operating environment and user interfaces experienced by managers or operators who provide and manage GII. Lead Body is proposed to be within SG4.
5) Advanced HCIs for telecommunication management. Q3/10 is proposed to become Lead Body.
6) Software architectures for advanced HCIs. Q3/10 is proposed to become Lead Body.

Question 1/10: Description techniques for GII interfaces (new)

Acting rapporteur:	Amardeo Sarma, Eurescom/ Deutsche Telekom

What are the needs for description techniques for interfaces to be standardised by ITU-T for the GII, and how can the description of interfaces across different studies related to GII be kept uniform?

Q1/10 is proposed to become Lead Body for the GII project Middleware for multimedia.

Question 2/10: ITU-T object definition language (new)

Rapporteur:	Joachim Fischer, Humboldt University/ Deutsche Telekom

Which activities should be done within ITU-T in order to influence the OMG IDL with requirements from the telecommunications applications domain? Could a future version of OMG IDL be recommended by the ITU-T as an object definition language for telecommunications systems?

Additional Tina-ODL features for telecommunications applications are:

1) Multiple interfaces and composition
2) Support of stream interfaces
3) Support of QoS.

A first list of potential issues for simplification and adoptions of SDL are:

a) Substitution of system, block and process by one entity
b) Introduction of an entity interface
c) Synchronous RPCs
d) Aggregation of entities
e) Introduction of a simple object-oriented data type concept
f) Introduction of a hierarchical state concept
g) Introduction of convenient programming constructs
h) Provide ITU-ODL – SDL mapping

Question 3/10: Software platforms and middlewares for the telecom domain (new)

Acting rapporteur: Arve Meisingset, Telenor AS

Which Recommendations should apply

1) in the area of software architectures and software platforms for the new generation of network and service management systems
2) for the role of middleware in the software framework to support the Global Information Infrastructure.

Q3/10 will in 1997 focus of collecting material from other organisations into a state-of-the-art report.

Q3/10 is proposed to become Lead Body for the GII project Advanced HCIs for telecommunication management.

Q3/10 is proposed to become Lead Body for the GII project Software architectures for advanced HCIs.

Question 4/10: Software quality of telecommunication system (revised)

Rapporteur: Benoit Hambenne, Belgacom

Which specific quality assurance activities should be recommended or tailored for specific telecommunication software and particular life-cycle process?

SG10 Plenary decided to initiate the Res. 1 approval procedure for a new draft Rec. Z.410, Quality Activities in Telecommunication Software Life-Cycle Processes, 6 May 1997.

Q4/10 will identify products and processes that are specific to telecommunication, such as IN, TMN and ATM, and tailor activities to these products and processes. This may include: rapid programming, use and reuse, re-sourcing, object-oriented programming, incremental programming, prototyping and acquirer involvement.

Recommendations:

Z.400 (03/93)	Structure and format of quality manuals for telecommunications software
Z.410 (05/97)	Quality activities in telecommunication software life-cycle processes.

Question 5/10: Specification of behaviour in GDMO (new)

Acting rapporteur: Anders Olsen, TeleDanmark

Considering that there is a need for specifying behaviour for GDMO and that initial requirements have been formulated as listed in Annex A to this Question and the experience so far that GDMO cannot simply be mapped onto the current SDL, which new Recommendations and changes to existing Recommendations, or other provisions are required for

1) specification of behaviour in GDMO
2) a combined use of GDMO and SDL based upon a behaviour specification in an SDL-like extension to GDMO
3) support of this combined use by integrated tools?

ISO has developed the GDMO+ language for stating behaviour in GDMO. Therefore, further work on this Question is dependent on expressed needs from SG4 and SG7.

Question 6/10: Maintenance and support of SDL (revised)

Rapporteur: Rick Reed, TSE Limited/UK

What new Recommendation and changes to existing Recommendations, or other provisions are required

1) for the release of a new Z.100 adapted to contemporary user requirements
2) to resolve the open items from the last study period
3) to support the use of SDL in new, emerging architectures and frameworks, such as ODP
4) to allow the use of SDL in combination with other methods and languages
5) for Z.110 maintenance
6) for Z.105 maintenance or alternatively integration of Z.105 with a new version of Z.100
7) for the maintenance of the methodology manual
8) to allow different natural languages and writing systems to be used with SDL to aid human understanding?

SDL’92 is a modern object-oriented specification and design language for real-time systems.

Z.105 provides common use of SDL with ASN.1. The SDL+ Methodology covers use of Z.100, Z.105 and Z.120.

Z.106 provides a common interchange format (CIF) for graphical SDL (SDL GR).

Telelogic and Verilog provide tool support for SDL.

Q6/10 is developing SDL-2000.

Q6/10 has issued a Request for proposals for a new Z.100 data model.

Q6/10 is Lead Body for Methodology within SG10.

Q6/10 is proposed to become Lead Body for the GII project Object-oriented environment.

Recommendations:

Z.100 (03/93)	CCITT specification and description language (SDL)
Z.100 App	SDL methodology guidelines
Z.105 (03/93)	SDL combined with ASN.1.

Question 7/10: Support for fast development of protocol standards using formal methods (new)

Acting rapporteur: Dieter Hogrefe, University of Lübeck/ Deutsche Telekom

Which new Recommendations, supplements or other provisions are required in order to be able to support fast development of protocol standards using formal methods?

A document is being developed on the use of SDL and MSC to support fast development of protocol standards.

Question 8/10: Testing based on formal specifications and validation of formal specifications (revised)

Rapporteur: Dieter Hogrefe, University of Lübeck/ Deutsche Telekom

Which new Recommendations or enhancements of existing Recommendations are needed to verify that formal specifications meet appropriate correctness criteria, e.g. absence of deadlocks and consistency, and how formal specifications can be utilised for conformance and interoperability testing, e.g. computer aided test case generation.

The work is carried out in collaboration with ETSI and use common text with ISO.

SG10 Plenary approved Rec. Z.500, Framework on Formal Methods in Conformance Testing, 6 May 1997.

Recommendations:
Z.500 (05/97) Framework on Formal Methods in Conformance Testing.

Question 9/10: Maintenance of message sequence charts (MSCs) syntax and semantics (revised)

Rapporteur: Øystein Haugen, Ericsson

What new Recommendations or enhancement of existing Recommendations, or other provisions are required in the area of Message Sequence Charts to

1) formalise the dynamic semantics and describe the static requirements of MSC-96
2) further extend and modify the language concepts as needed by users in both industry and standards bodies
3) correct minor errors and inconsistencies in Z.120
4) resolve the issues in the list of open items
5) improve the use of natural languages and writing systems with MSC to aid human understanding?

MSC-96 is a specification language for message sequences with rich structuring concepts.

Due to late translation in ITU TSB, MSC-96 exists in Korean only, in addition to the non-official English original.

Focus points for MSC-2000 are: Control logic, decomposition, multicast and synchronous messages, and real time constructs. Also, a common data model and a common methodology will be developed together with Q6/10.

Q9/10 will attempt to influence OMG to use MSC as a part of its UML language.

SG10 Plenary determined to initiate the Res. 1 approval procedure for draft Rec. 120 Annex B, Structured Operational Semantics for MSC-96, 6 May 1997.

Recommendations:
Z.120 (09/96) Message sequence chart (MSC).

Question 10/10: Maintenance and evolution of CHILL (revised)

Rapporteur: Juergen Winkler, University of Jena/ Siemens

Which Recommendations should apply to the maintenance of ITU-T Recommendation Z.200 and/or related documents with respect to

a) introduction of overloading
b) possible harmonisations between ASN.1, SDL, and CHILL
c) introduction of elements for distributed programs
d) improvement of piecewise programming
e) introduction of persistent data
f) use of different natural languages and writing systems with CHILL to aid human understanding.

CHILL-96 is a modern object-oriented language for the development of safe and maintainable code for real-time systems.

Siemens, ETRI and Kvatro provide CHILL compilers and tool support.

Focus points for extensions of CHILL are: Overloading, multiple inheritance, distributed programs, friends and extended character sets.

Q10/10 is Lead Body for extended character sets within SG10.

A new version of CHILL is planned to be approved in April 2000.

Recommendations:
Z.200 (11/93) CCITT High Level Language (CHILL).

Question 11/10:
Graphic GDMO (revised)

Rapporteur: Arve Meisingset, Telenor AS

Which Recommendations are required to provide a graphic formalism of GDMO and guidelines for its usage?

SG10 Plenary approved Rec. Z.360, Graphic GDMO, 6 May 1997.

Q11/10 will attempt to harmonise Graphic GDMO and UML notations.

Recommendations:
Z.360 (05/97) Graphic GDMO.

Question 12/10:
Specification of HMI data for a GDMO/ASN.1 object model (revised)

Rapporteur: Arve Meisingset, Telenor AS

Which Recommendation is required to

1) define a minimum set of HMI data needed to provide an HMI implementation with the national language specific labels, online help, and rules for the presentation of the various elements of data defined by a GDMO/ASN.1 object model?
2) provide a formalism which enables specification of these HMI data for particular GDMO/ASN.1 object models?

Requirements and a framework for the Question are developed. The Question is terminated, due to lack of resources.

Recommendations:
Z.351-Z.352 (03/93) Data oriented human-machine interface specification technique.

Question 13/10:
Design principles for human-machine interfaces (HMI) for the management of telecommunications network resources and services (new)

Rapporteur: Tom Winlow, Nortel

Which Recommendations are required to: provide a comprehensive set of design principles which can provide guidance to HMI designers?

Determination to initiate the Res. 1 approval procedure for new Rec. Z.xxx is planned for March 1998.

Recommendations:
Z.301-Z.341 (1988) CCITT man-machine language.

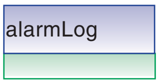
Arve Meisingset is Research Scientist at Telenor R&D, Kjeller.

email address:
arve.meisingset@fou.telenor.no

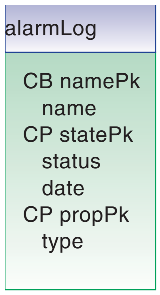
Graphic GDMO

ARVE MEISINGSET

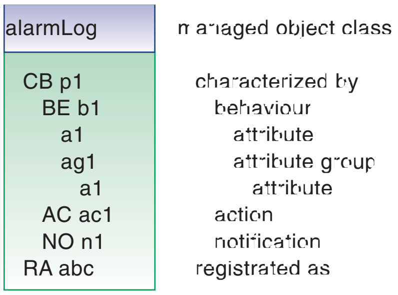
Managed Object Classes



1a. Managed Object Class labels are adjusted to the left and separated from the rest by a horizontal line

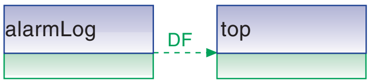


1b. Characterised By and Conditional Packages are indented

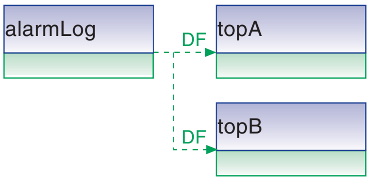


1c. Packages, Behaviour, Attributes, Attribute Groups, Actions, Notifications and Registered As are shown by indentation

Derived From

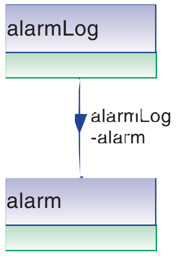


2a. Derived From is depicted by one or more dashed arrows supported by the DF label

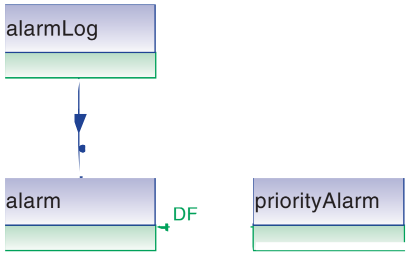


2b. Multiple inheritance by Derived From

Name Binding



3a. Name Binding is indicated by a filled triangle pointing towards the subordinate Managed Object Class



3b. Name Binding to And Subclasses can be indicated by a filled dot both for superior and subordinate Managed Object Classes. The example shows And Subclasses for subordinates only

ITU-T Recommendation Z.360, Graphic GDMO, provides a graphic notation for a subset of the information provided in the templates defined in Recommendation X.722 Open Systems Interconnection – Structure of Management Information – Part 4: Guidelines for the definition of Managed Objects.

The purpose of the graphic notation is to provide an overview of TMN specifications for TMN specifiers, implementors and users. The graphic notation provides an overview of what definitions exist and the relationships between these definitions.

To accomplish this overview, not all information provided in the templates has to be presented in the graphic notation. Therefore, the graphic notation covers only a subset of the information found in and required by the templates. Typically, a graph may depict all information in one Recommendation. However, the specifier is free to include and exclude what he finds convenient, graphs can overlap, and there is no Recommendation on what shall be included in one graph or not.

Telenor has been the prime contributor to the development of the new Recommendation Z.360, Graphic notation for GDMO. Recommendation X.722, Guidelines for the Definition of Managed Objects, is an alphanumeric notation for the definition of data for TMN. TMN is defined in Recommendation M.3010, Principles for a Telecommunications Management Network. GDMO is an object oriented notation. However, specifications using GDMO are hard to overview and informal graphs and texts are misleading as to the contents of the specifications. Therefore, the primary requirements to the new graphic notation for GDMO have been:

1. Alphanumeric GDMO/GRM specifications shall have priority over graphs using Graphic GDMO
2. The Graphic GDMO shall be true to the Alphanumeric GDMO/GRM.

GRM refers to Recommendation X.725, General Relationship Model, which can be considered to make up an extension to GDMO. Data value syntax is defined by using ASN.1, Recommendations X.208 or X.680, Abstract Syntax Notation One.

There are several popular graphic notations for the definition of data available in the market. However, Graphic GDMO is the only graphic notation which

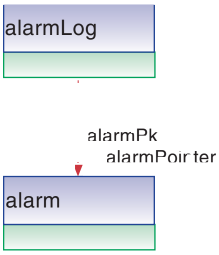
- satisfies the second requirement
- depicts a large set of details from the GDMO specifications
- provides the most compact overview of GDMO specifications.

In this paper we present some aspects of Graphic GDMO. For a full exposition, see ITU-T Recommendation Z.360.

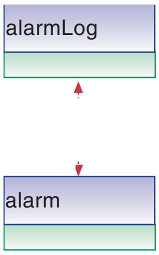
Recommendation M.3100, Generic Network Information Model, has been used to test the usability of Graphic GDMO.

Note: Rec. Z.360 does not prescribe colours. Therefore, the colours used in this document are optional.

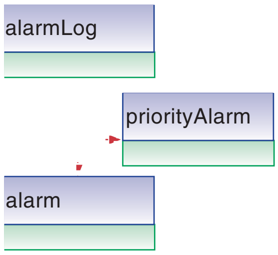
References



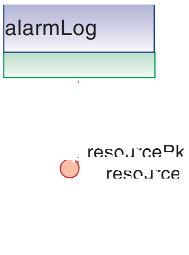
4a. References by pointer attributes can be depicted by dotted arrows, labelled <pk, attr>



4b. Two mutually dependent references are depicted by a two-way dotted arrow

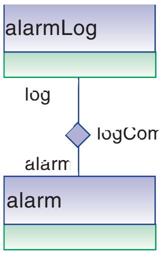


4c. A reference to several alternative Managed Object Classes is depicted by branching

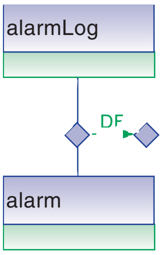


4d. A reference to any object class is depicted by an arrow with a round head

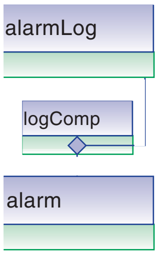
Relationships



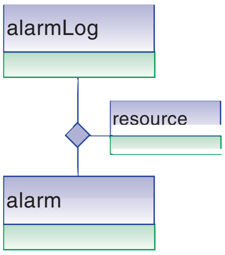
5a. Relationships are indicated by diamonds, Roles by role labels



5b. Relationships can be Derived From other relationships

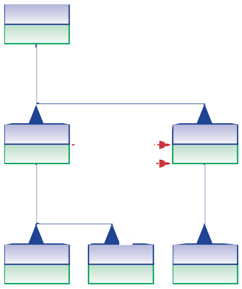


5c. Relationships can be reinterpreted as Managed Object Classes

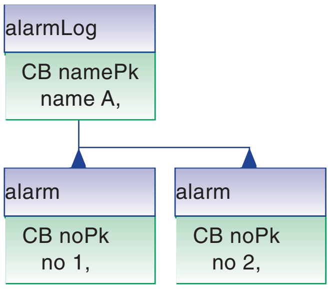


5d. Relationships can be n-ary

Instances



6a. Instance graph



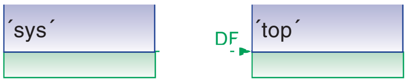
6b. Values are followed by commas

Instantiation graphs are provided by:

1. Removal of Derived From
2. Removal of alternative Name Bindings
3. Instantiation

Instance symbols are identical to class symbols. Name Bindings are replaced by subordination, depicted by reversed arrow heads. Values are indented and followed by comma.

Long labels



7a. Abbreviations are placed between single quotation marks

‘sys’:	system
‘top’:	“Rec. X.721 I ISO/IEC 10165-2”: top

Realistic example from Rec. M.3100

'top':
Recommendation X.721: 1992':top

'organization':
CCITT X.521':
organization

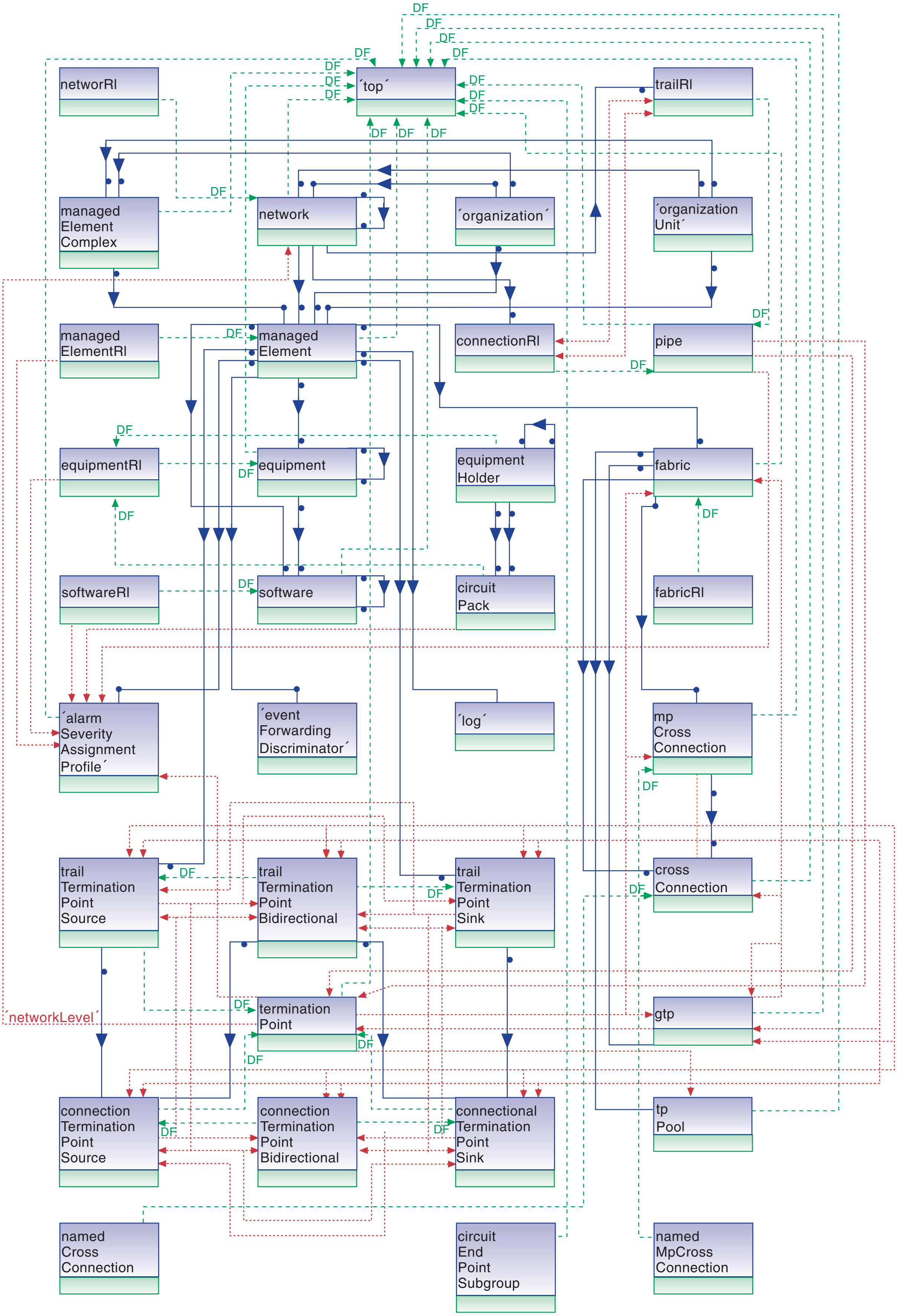
'organizationUnit':
CCITT X.521':
organizationUnit

'alarmSeverityAssignmentProfile':
Recommendation X.721: 1992':alarm-SeverityAssignment-Profile

'eventForwardingDiscriminator':
Recommendation X.721: 1992':event-ForwardingDiscriminator

'log':
Recommendation X.721: 1992':log

'networkLevel':
networkLevel-Package, network-LevelPointer



For a presentation of the author, see page 93.