

Wide-Area Publish/Subscribe Service Discovery – Application to Personal Networks

WASSEF LOUATI, DJAMAL ZEGHLACHE



Wassef Louati is a PhD student at INT, Evry, France

This paper explores the use of event-based communications to design publish/subscribe service discovery architectures that are adequate for wide-area pervasive environments. Some service discovery protocols and paradigms lack this event-based communication capability. They also often present weaknesses in expressiveness, extensibility and flexibility. Some service discovery frameworks include eventing but are limited to local area discovery. All these features and capabilities are needed to create the next generation of mobile and large-scale Internet services. A Wide-area Publish/subscribe Service Discovery (WPSD) framework meeting such requirements is presented and analyzed in terms of design and implementation characteristics. The applicability of WPSD to service discovery and management in Personal Networks is also examined.



Djamel Zeghlache is Professor and Head of Wireless Networks and Multimedia Services Dept., INT, Evry, France

1 Introduction

Service discovery allows services to advertise themselves so that clients can query and possibly access them. Service discovery is most often local and hence limited in range. With the distribution of services in increasingly large-scale environments, new paradigms for service discovery, creation and provisioning are needed to cover wider areas and prepare for this evolution. Wide-area service discovery has become a requirement and has consequently attracted much interest¹⁾.

Wide-area service discovery is a key enabler for wide-area pervasive environment and especially for *Personal Networks* (PN) that have gained interest recently in the research community [1]. The PN concept extends the Personal Area Network (PAN) by including *remote* personal devices in the network architecture (Figure 1). The PN can be viewed as a collection of geographically scattered clusters that can communicate via interconnecting structures or in ad hoc mode. A PN cluster is a network of devices characterized by a common trust relationship and

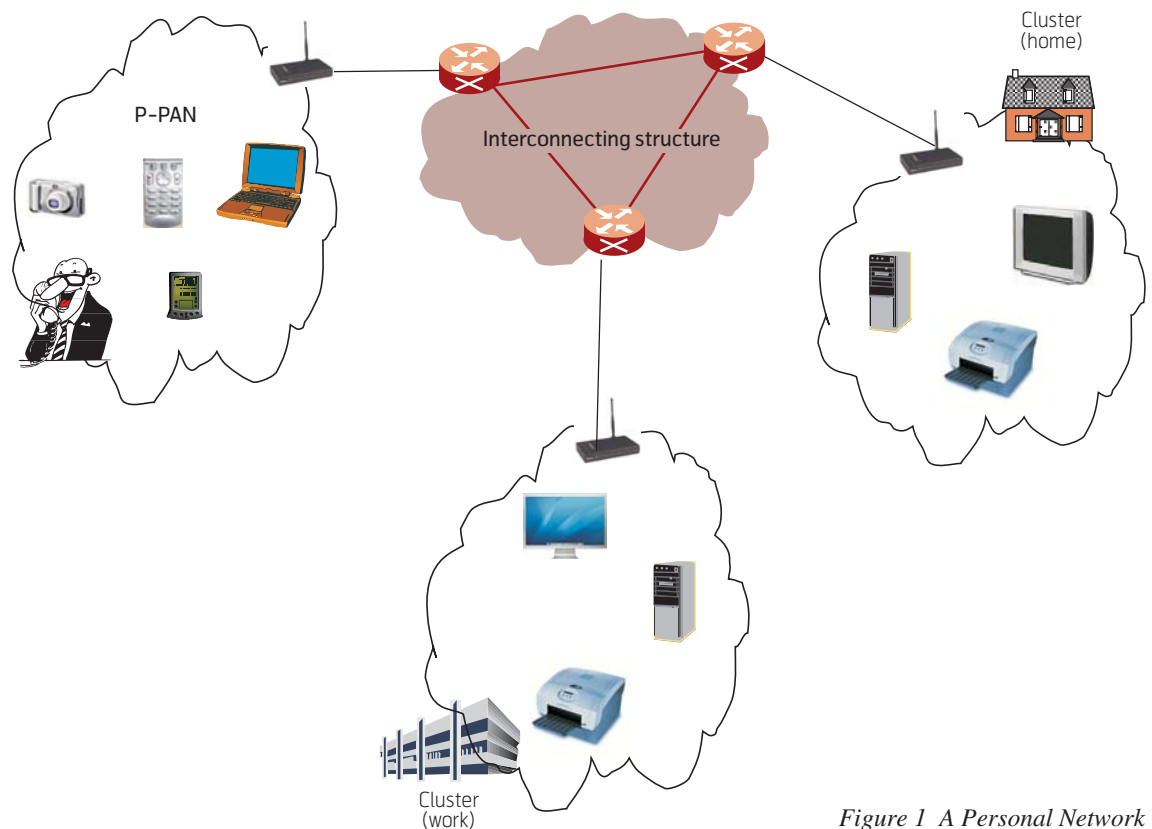


Figure 1 A Personal Network

¹⁾ The wide-area service discovery is the current focus of the IRTF P2PRG Content, Resource and Service Discovery (CORE) Subgroup. The purpose of this subgroup activity is to define a research agenda within the P2PRG community to evaluate existing research, identify requirements, and develop solutions for wide-area P2P content, resource, and service discovery.

located within a limited geographical area (e.g. home cluster, office cluster, etc.). The PAN surrounding the user is a special cluster called Private PAN (P-PAN).

Wide-area service discovery can support many applications including service control, dynamic service creation, context awareness and network management. Personal networks can also considerably benefit from wide-area service discovery frameworks.

Clients (e.g. PN user, PN application, PN management system, etc.) need in fact to be notified about services that match their queries at service announcement times. Clients may also require to be notified about modification of state and access information of services due to mobility and dynamic changes. To perform such notifications, service discovery should embed *eventing* mechanisms.

Existing architectures supporting eventing (e.g. UPnP [2], Jini [3]) are suitable for small-area and local discovery but cannot support wide-area and remote discovery because they use synchronous or multicast communications that suffer from scalability problems.

In the context of wide-area environments, eventing is known as *event-based communication* or as a *publish/subscribe* model. In this model, subscribers describe the events (data, messages, etc.) that they want to receive from publishers, without having previous knowledge of each other's locations. The model ensures the event flow from publishers to subscribers so that notifications can be received by the subscribers. The strength of this model is its loose coupling between publishers and subscribers in time, space and synchronization. This characteristic makes event-based communication a scalable communication model that is widely used in many large-scale Internet services and mobile computing environments. A natural evolution for service discovery is thus to integrate event-based communication to provide eventing capability and achieve wide-area service discovery.

The objective of this paper is to address such service discovery frameworks and especially analyze a *Wide-area Publish/subscribe Service Discovery* (named WPSD) framework as a potential solution.

Many applications, especially PN applications and services, can benefit from WPSD. These applications need to be informed immediately about changes in context data, services states and PN networking and management.

The paper conducts its analysis of wide-area publish/subscribe service discovery by first presenting an

overview of event-based middleware. This description is followed by definitions and requirements on the proposed WPSD service discovery framework. A general architecture for WPSD is then presented and a specific implementation reported. Finally, the applicability of WPSD to Personal Networks is examined.

2 Event-Based Middleware

An event-based middleware provides publish/subscribe communication between components of large-scale distributed systems. In an event-based middleware, events represent the basic or fundamental communication mechanism [4].

Event consumers express their interest, in receiving events, in the form of an event *subscription*. Event consumers are commonly called *Subscribers*. Event producers produce events, which will be delivered to all interested Subscribers. Event producers are commonly called *Publishers*. Thus, an event-based system has an asynchronous many-to-many communication model, where the Publishers and the Subscribers are decoupled. The event delivery is performed by *event brokers* that form an overlay network that routes the events from the Publishers to the Subscribers. The path is set up by *advertisements* and subscriptions sent by Publishers and Subscribers respectively.

The event-based middleware architecture has four layers (Figure 2) [4][5]. The lowest layer is the network layer. On top of this layer, an overlay network implements application-level routing (content-based, peer-to-peer (P2P), etc.). The overlay handles the routing between the application-level nodes (event brokers, peer nodes, etc.) and has a self-organizing mechanism that supports the node dynamic changes (when joining and leaving the overlay).

The event-based middleware layer provides the programmer an interface to handle the events (advertisement, subscription, publication, etc.) and manages also the storage, event filtering and notification operations in the event brokers.

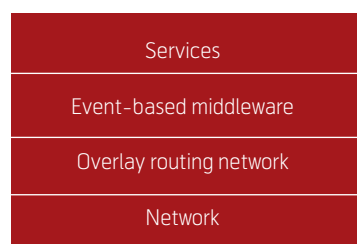


Figure 2 Event-based middleware architecture

The event-based middleware layer can benefit from extensions offered by the service layer that can provide additional services such as security and QoS. An event-based system can be extended to support QoS-aware paths between the Publishers and the Subscribers such as in [6]. Introducing security into the system can be achieved for example by making the event brokers OASIS-capable (Open Architecture for Secure Interworking Services) [7] to perform a boundary policy-based access control to the middleware. The broker would use the OASIS policy to decide if the advertisement/subscription can be admitted [8].

3 Wide-Area Publish/Subscribe Service Discovery

Event-based middleware has a layered architecture designed to offer multiple services. Service discovery can be one possible offered service. Even though some event-based systems can be tailored for service discovery, most adaptations suffer from significant overhead. The event content must be adapted to a service description and rendezvous brokers must be previously set up by the advertisement to perform notification. The best solution, as adopted by WPSD proposed in this paper, is to build an event-based system designed for service discovery from the start. The event content would be explicitly defined (service information) and the routing of events between the Publishers and the Subscribers carried out at service (event) delivery time.

3.1 Design Requirements

To design a Wide-area Publish/Subscribe Service Discovery (WPSD) system, we define a set of system requirements resulting from a mix of our experience in wide-area service discovery [9][10], the event-based middleware requirements of [4], event-based properties [11] and the problem statement specified recently by the P2PRG CORE subgroup in [12]. The requirements consist of:

Multiple event support

A WPSD system should notify the subscribers when new events occur:

- Appearance of new services
- Departure of services
- Dynamic changes in service state
- Changes in mobile service location
- etc.

Reliability

A WPSD system must be resilient to entity failures. The failure of a part of the system must not cause total system failure. The system must be always

aware of the location of the nomadic services and their clients, especially with the emergence of wireless technologies. The system should also support the mobility of the brokers.

Scalability

The scalability is an important requirement for any WPSD system since the number of involved devices, users and services is in continuous increase.

Dynamic update

In a WPSD system, a subscriber should not receive an interrupted stream from the publisher. When a service is dynamic and/or mobile, its service information update should not be performed by removing the old information and announcing the new one. Only an update due to the new information should take place.

Security

A WPSD system should support security and privacy protection. Especially, the security mechanisms should protect the system from attacks related to the distributed architectures.

Interoperability

Many service discovery technologies can be used at the edge of a WPSD system. Thus, WPSD should be able to interact with different service discovery systems. This leads to the following interoperability requirements:

- A linking mechanism (i.e. gateway) should be used to make the translation between the WPSD and the local discovery systems when service description semantics differ.
- Gateways at the edges should have an API that supports all operations (addition, removal, dynamic update, security, etc) to enable interaction.
- The WPSD system should use an expressive and extensible service description language to be capable of translating descriptions to/from other languages.

There is a trade-off to achieve between expressiveness in the service description language and scalability [13]. A highly expressive service description language would require more processing and would make the system less scalable. The presented design in this paper takes into account this trade-off by striving for a balance between expressiveness and scalability.

3.2 Design Decisions

In this section, we describe how to integrate the publish/subscribe model in a wide-area service discovery

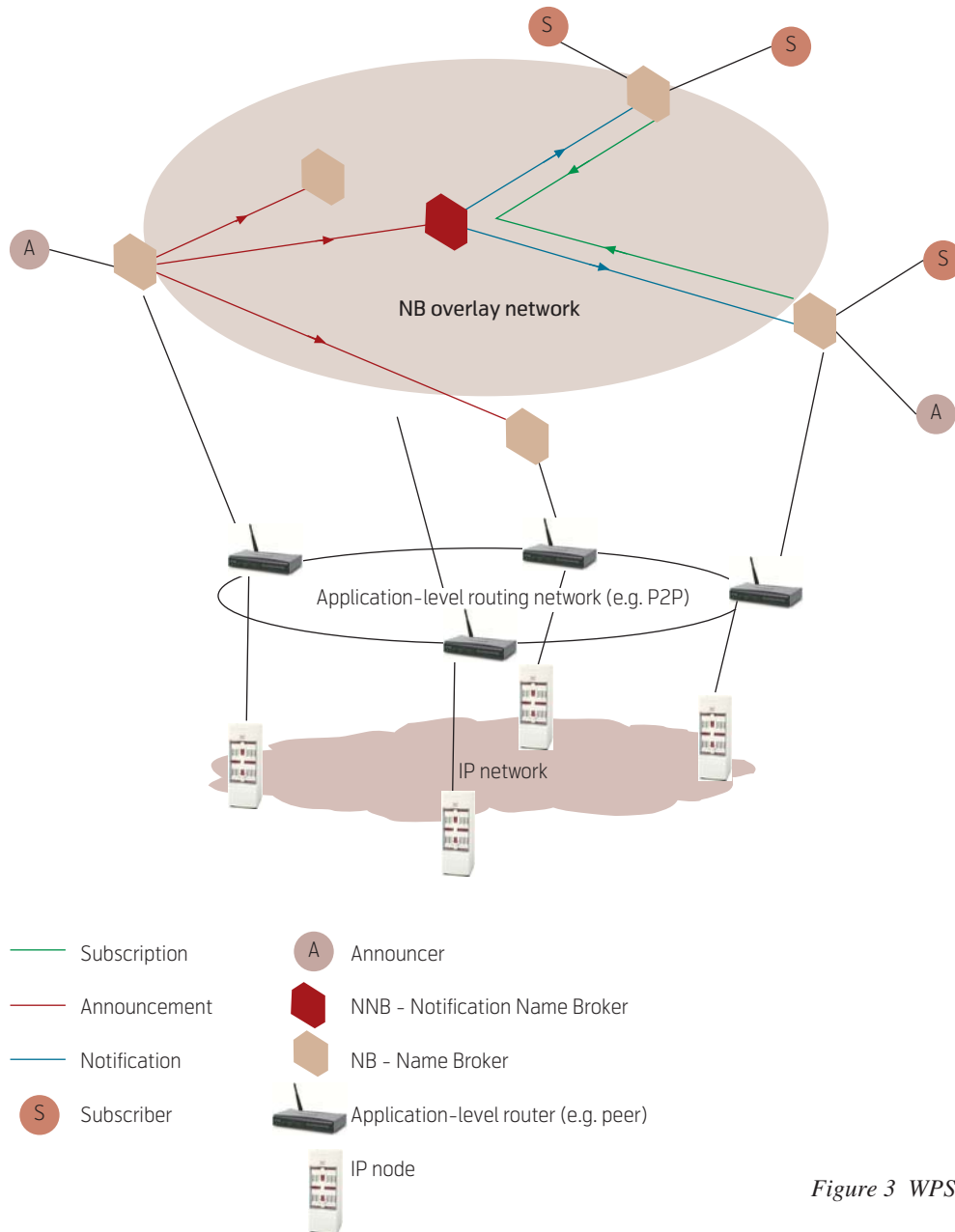


Figure 3 WPSD architecture

leading to a WPSD system. In this description, we define the required additional operations, the involved entities and their interactions.

To start with, we must select the right model that should be applied to service discovery. Mainly, there are two categories of publish/subscribe models: topic-based and content-based. Each form has its own manner of expressing the event subscriptions. The topic-based model uses a predefined set of topics or subjects to identify the events. In content-based model, the events do not have such restricted structure.

In topic-based model, the topics are not expressive. They just use predefined labels and this is not sufficiently rich to specify service subscriptions. In the content-based model, a service subscriber can make more specific service subscription using richer description formats (e.g. attribute/value pairs, XML,

etc.). WPSD hence adopts the content-based model since it is more suitable for service discovery applications.

In a WPSD system, a service announces its *service information* in the form of a pair (*name, name-record*). The name denotes the service description (e.g. attribute/value pairs) and the name-record denotes the access information to the provider of the service, such as its IP address. By handling name to address mapping, a WPSD system acts also as a naming and name resolution system.

The producer and consumer in a WPSD system are the service *Announcer* and the service *Subscriber*. The event brokers of WPSD, called *Name Brokers* (NB), are *name information databases*. The NBs *route* names from the Announcers to the Subscribers using a *name-based routing* paradigm. The routing

occurs on an application-level routing that can be achieved using a P2P network or any other type of overlay (Figure 3).

The Subscriber announces a *subscription name* to perform one of the three subscription types below:

- *Name subscription* to be notified about names that match *exactly* all the subscription name attributes;
- *Supername subscription* to be notified about *any* name that match the subscription name;
- *Name-record subscription* to be notified about any modification of the name-record information (due to Announcer mobility or dynamic changes in environment) of the subscription name.

The name subscription is performed using the usual name announcement mechanism of the current service discovery architecture. The name-record of a subscription name is called *subscriber name-record*. The *Notifications* are performed after name information database updates by the Announcers (addition, removal, and dynamic update). For a given name update session, an NB holding the name database changes is called *Notification Name Broker* (NNB). The location and the number of NNBs for a given subscription depend on the used name-based routing of the current service discovery protocol. The notifications of the name update are sent by an NNB to the appropriate Subscribers identified by their name-records.

For example, suppose Alice wants to be notified about all her cameras (in her home, office, etc.) and dynamic changes in their states. Alice submits from her Subscriber a supername subscription like [service = camera] [owner = Alice [cluster = *]]. One or more NBs that receive the supername will play the role of NNBs. When a new camera is added or the states of existing cameras change, the Announcers related to the cameras announce or update the names. The NNBs notify Alice's Subscriber about these new changes.

WPSD introduces the ability to make dynamic updates of the announced name information. When Announcers or Subscribers move, their access information (i.e. name-records) is updated in the NBs. The corresponding NNBs notify the Subscribers so that they are always aware of the Announcers' locations.

3.3 WPSD Scalability

WPSD is intrinsically scalable since Announcers and Subscribers are decoupled, as in any publish/subscribe system. Many other factors improve the

scalability of WPSD. The NBs self-configure into a distributed application-level overlay network to perform name-based routing. By integrating naming and routing in the application-level, the scalability can be maintained while highly expressive names are used.

The scalability is also improved by the load balancing ensured by WPSD. The load induced by subscription and announcement is quite balanced and reduces the risk of an NB becoming a bottleneck node in the network. This is due to the NB location that depends on the name to be routed. Since the NNB location depends on the subscription names, the notifications are issued from different NNBs. This distributes naturally the load and avoids congestion in the NB network.

Scalability and load balancing occur naturally when a Distributed Hash Table (DHT) is used as the application-level routing network in the WPSD system. DHTs, such as Chord [14], CAN [15], Pastry [16] and Tapestry [17], create structured P2P networks that, given a key, can efficiently lookup the peer at which the corresponding value is stored. The DHT storage and lookup processes scale logarithmically with the number of peers in the network. For DHT-based designs, the NB location may be determined by hashing names (announced or subscription names) and looking up the NB peers responsible for the resulting keys (i.e. the hash values). This will be illustrated with more details in section 3.6.

3.4 WPSD Security

In [18] the authors present a number of security issues and requirements in wide-area publish/subscribe systems. Based mainly on this analysis, several solutions have been proposed to secure publish/subscribe systems. Some of these solutions can be adopted and added to secure a WPSD system.

Three main security mechanisms should be addressed. First, an *access control* mechanism should be designed at the boundary of the NB network. Only authenticated and authorized Announcers and Subscribers should be able to access the NB network. To address access control, the authors in [19] present an access control mechanism based on publication/subscription message content. This is a suitable solution for WPSD since it is a content-based publish/subscribe model.

Second, a mutual *trust* should be provided between the Subscribers and the Announcers. The most advanced solution, proposed by the authors in [20], is to introduce the notion of scoping for structuring publish/subscribe systems. Scopes delimit groups of Publishers and Subscribers on the application level and

control the dissemination of notifications within the broker network. Hence, they are suitable for building groups of trust (i.e. groups whose members belong to the same authentication domain). The trust relationships in scopes are established using public key infrastructures and access control methodologies. WPSD can adopt this solution as well, especially when the NBs belong to different administrative domains.

Finally, *confidentiality* and *integrity* of the announced names, the subscription names, and the inter-broker messages should be provided. The common solution is to encrypt and sign the messages, as has been achieved by the work in [8] and [20].

3.5 The Support of the WPSD Model

The integration of the publish/subscribe model into an existing wide-area service discovery system can lead to a WPSD system. To support this integration, the selected system should be flexible, extensible and scalable. Several architectures can support the integration of the WPSD model as long as their extensions fulfill all previously mentioned requirements. In fact, this depends mainly on the *infrastructure model* of the service discovery protocol. Wide-area service discovery protocols ([21]–[25]) have normally a *structured directory²⁾-based* infrastructure model [26].

There are two major existing directory structures for wide-area service discovery: *flat* and *hierarchical* structures. In a flat structure, exemplified by protocols such as INS/Twine [21] and Superstring [22], directories have P2P relationships. Hierarchical structures, including SSDS [23], CSP [24] and GloServ [25] have a DNS-like directory structure.

Hierarchical structures overcome the overloading problem of directories by spawning the load on the child nodes. Nevertheless, they cannot scale easily to WPSD integration. The announcements and the notifications would propagate up and down through the hierarchy and can turn the root node into a bottleneck. Even if several hierarchies may co-exist, the hierarchies would not handle efficiently subscription names that include orthogonal attributes. For example, imagine a WPSD system having two hierarchies of directories: one based on service location and the other based on the service property. A user may subscribe to its own services located in a given geographical area. The hierarchy described would not handle this subscription in a scalable manner, since both hierarchies will be involved in the notification.

Service discovery protocols having flat structures generally rely on DHT. These protocols inherit DHT scalability, efficiency, reliability and robustness. Hence, DHT-based service discovery protocols are more suitable for WPSD integration than hierarchical approaches. The next section describes an implementation example of a WPSD system using INS/Twine, a typical DHT P2P-based service discovery protocol.

3.6 Implementation Example: WPSD Based on DHT P2P-Based Service Discovery

A viable framework to implement a WPSD system is INS/Twine [21] that attracted our interest for its flexibility, expressiveness and ability to use any kind of identities to describe objects, nodes, services while providing information on their location. This system resolves names or identities into IP addresses for networking purposes.

INS/Twine is a P2P-based service discovery protocol with an expressive, responsive and robust system designed for dynamic and mobile environments. Services are described using a hierarchical attribute-value pair naming scheme such as XML. The description is called intentional name. The INS/Twine architecture consists of a network of Intentional Name Resolvers (INRs) that provide name resolution and name-based routing. The service providers and the clients are located at the edge of the INR network.

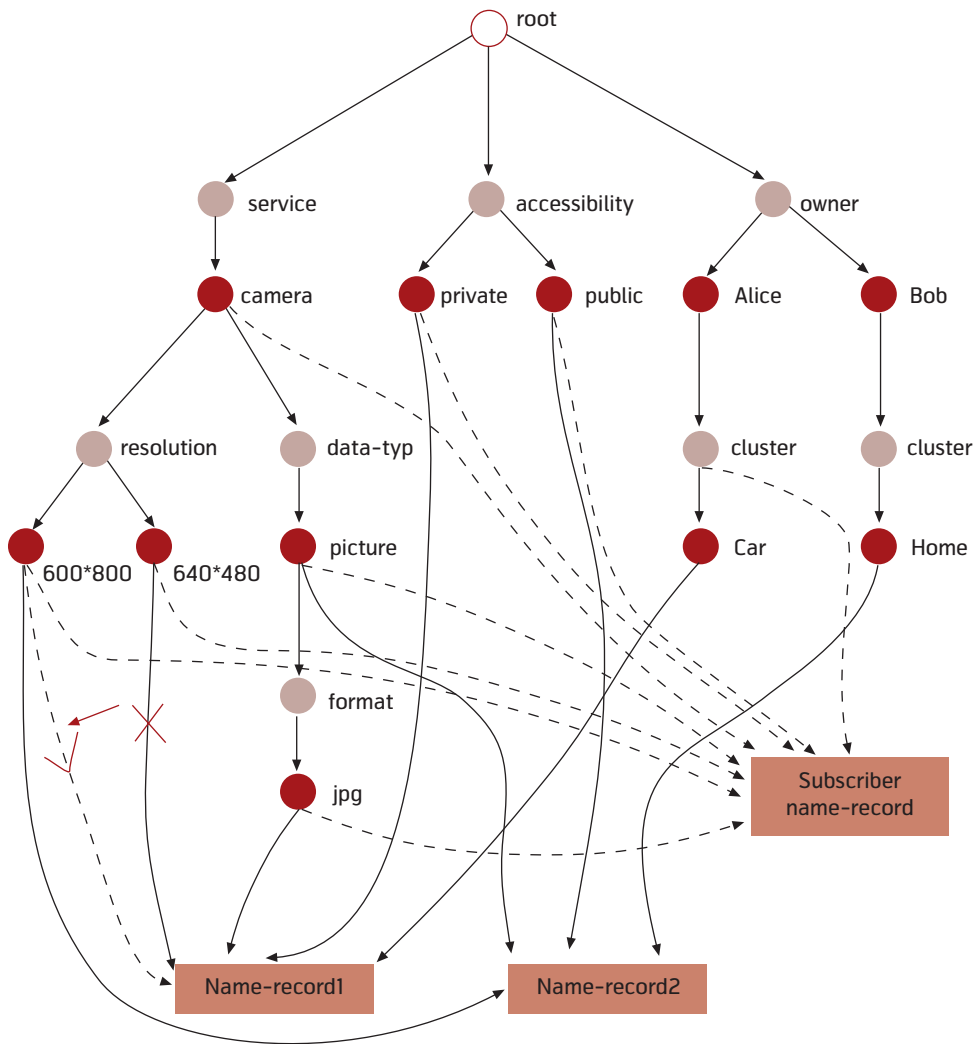
The INR name database has a data structure called a name-tree. The INRs self-configure into an application-level distributed overlay network over a DHT network to disseminate and route name information to the appropriate locations.

INS/Twine performs hash-based partitioning of service information among a subset of INRs. The querying has a similar mechanism. INS/Twine extracts each unique prefix subsequence of attributes and values from a name (an announced name or a name query). Each subsequence, called a strand, is then used to produce a separate key. INS/Twine uses a DHT process to map the keys to appropriate resolvers into which the service information or query is forwarded.

WPSD can be designed by extending the INRs of the INS/Twine framework. The resulting INRs play the role of Name Brokers (NBs). The Subscribers correspond to clients and the Announcers to service providers.

For name announcement, the name is added in the name-tree of the corresponding NBs. This is per-

²⁾ The directory, also often called broker or resolver, is the entity that stores service information and processes announcements and queries.



Name-record1: of Alice name ([service=camera[resolution=600*800][data-type=picture[format=jpg]][accessibility=private][owner=Alice[cluster=car]])

Name-record2: of Bob name ([service=camera[resolution=600*800][data-type=picture]][accessibility=public][owner=Bob[cluster=Home]])

Subscriber Name-record: of Alice surname subscription ([service=camera][accessibility=*][owner=Alice[cluster=*]])

Figure 4 Name database management in INS/Twine-based WPSD

formed by adding the corresponding name-record to all leaf values of the name in the name-tree (see Figure 4).

The dynamic update of a name and its access information consist of finding the corresponding name-record in the name-tree. If the dynamic update concerns the access information of the name, the update affects the name-record itself. If the name needs to be updated (i.e. updating a given value in the name), the name-record is detached from the old value and associated to the new one.

For name subscription, the subscriber name-record will be associated to the leaf values of the subscrip-

tion name. In the case of the surname subscription the association will involve all the surname children values. The name-records of a matching name or surname will be associated to the values having the subscriber name-record. This association will trigger the notification action. The detachment of an association, due to name removal or update, will also trigger the notification.

The notification can be achieved by any NB corresponding to a key computed from the strands of a subscription name. To achieve scalability, one single NB is selected to play the role of the NNB. This NNB corresponds to the longest strand of the subscription name.

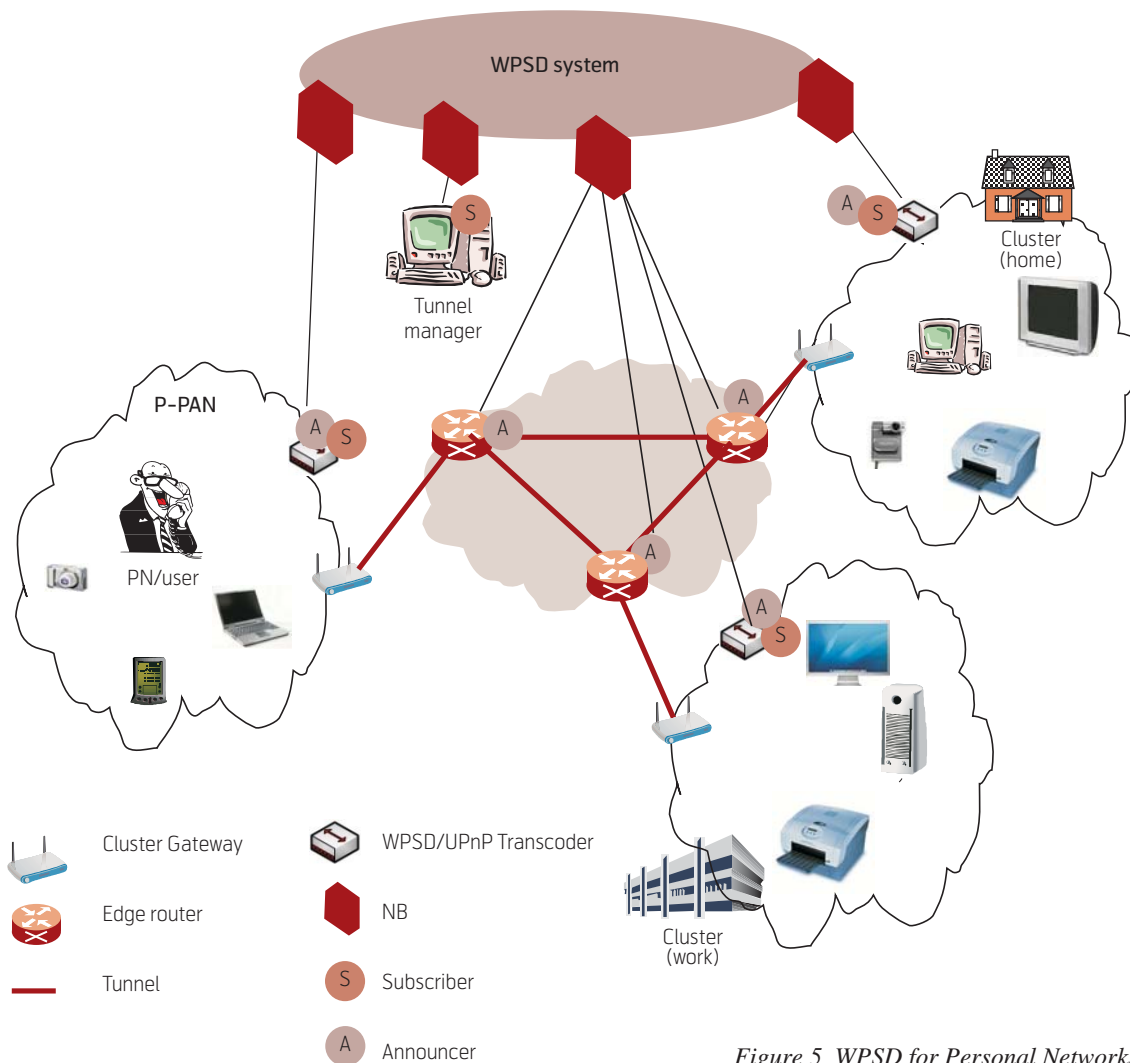


Figure 5 WPSD for Personal Networks

Figure 4 gives an example of a name-tree management. The figure illustrates a dynamic update, a supname subscription and two announced names (related to the two depicted name-records). For the dynamic update, the resolution attribute of the camera service has been updated from a camera resolution value of 640*480 to the 600*800 value.

The name-tree depicts a supname subscription ([service = camera][accessibility = *][owner = Alice [cluster = *]]). The supname is composed of four strands: (service, camera), (accessibility), (owner, Alice) and (owner, Alice, cluster). The subscriber will be notified of Alice's name as the corresponding name-record is associated to the values having the subscriber name-record. The NNB corresponds to the fourth strand, as it is the longest one.

4 WPSD Applicability for Personal Networks

In this section, we will show the applicability of the WPSD system for Personal Networks (PNs) (Figure 5) as a communications middleware between local

service discovery frameworks with eventing capabilities. WPSD can also assist PN establishment and context information delivery.

4.1 WPSD-Based PN Establishment

PN networking entails the establishment of secure inter-cluster communication using tunnels. To establish tunnels between clusters, one needs to be capable of locating the clusters and their points of attachment to intermediate networks. To provide this information, the concept of a PN Agent has been introduced [27] to assist PN establishment. A possible approach is to use tunnel managers to send tunnel configuration parameters to tunnel end-points needed to connect distant PN clusters. Tunnel establishment based on policies relying on names is referred to as *name-based tunnel establishment* [28][29]. A scalable publish/subscribe naming system such as WPSD can enable PN establishment by embedding the PN Agent.

This can be achieved as follows. The tunnel end points, depicted in Figure 5 as access routers, shall include the WPSD Announcers. The Announcers

declare the names of the attached clusters. The cluster names include the PN identifier and the cluster identifiers (e.g. [PN = Bob [Cluster = Home]]). The PN Agent maintains information about the cluster names and their attachment points (the IP addresses of the access routers are in the name-records).

The tunnel management, using an embedded WPSD Subscriber, performs cluster name subscription and subscribes to the name-record of each registered cluster name to become aware of the cluster dynamics and mobility induced changes.

When a cluster changes its attachment point, the cluster name is removed from the Announcer in the old tunnel end-point and is announced by the prospective new tunnel end-point. Once the Subscriber is notified of these changes (by obtaining the IP addresses of the old and new tunnel end-points), the tunnel manager sets up the new tunnel and tears down the old one.

Clusters can frequently merge or split. For example the home cluster splits into the P-PAN and the home cluster. The P-PAN can merge with other clusters, e.g. office. During these dynamic changes, the Announcers in tunnel end points can perform updates of the cluster identifier attribute value. The advantage here is that the update can occur without completely removing or announcing again the cluster name. The active tunnels can be maintained thus avoiding additional tunnel tear downs and set ups.

4.2 UPnP Extension for Wide-Area Service Discovery Using WPSD

Universal Plug and Play (UPnP) [2] is a widely deployed local service discovery protocol. WPSD can be used as communication middleware between UPnP-enabled PN clusters. With this combination, a PN user can access, from any location, any UPnP service in any given UPnP-enabled PN cluster.

This interoperability requires the design of interworking components called Transcoders. The Transcoder translates UPnP service descriptions into WPSD names and vice versa. The Transcoder binds also the UPnP operations (announcement, discovery, and eventing) into WPSD operations and vice versa.

By using WPSD, the discovery can benefit from the most recent dynamic updates of the service descriptions. The UPnP Client of the Transcoder can subscribe to all UPnP services in its associated cluster using the UPnP eventing functionality. When receiving the updated service description from UPnP, the WPSD Announcer of the Transcoder can infer the updated attributes and achieve the value updates. Hence, the UPnP service database remains always up to date.

In addition, UPnP eventing can be concatenated to WPSD eventing if the UPnP services cannot be directly accessed remotely. The distant nodes would subscribe to the remote UPnP service through WPSD.

UPnP is just an example to illustrate the interoperability benefits of WPSD. WPSD can cooperate with other local area service discovery supporting the eventing function such as Jini [3].

Interoperability of WPSD with popular service discovery protocols and frameworks should ease its large-scale deployment and adoption as a wide-area service discovery architecture.

4.3 Context-Awareness Using WPSD

The publish/subscribe feature of WPSD can enable context awareness in Personal Networks if nodes subscribe to context information services offered by nodes acting as context sources. It would be sufficient to use a name Announcer that publishes names that correspond to key context information advertised by the context sources. The context data consumers would build from these names a subscription name and simply subscribe to automatically receive notifications about changes in context from the WPSD system. Services and applications can thus be made context aware and can adapt according to dynamic changes in the overall PN architecture and constituents.

5 Conclusion

This paper examined the introduction of publish/subscribe paradigms into current service discovery frameworks to achieve flexible, extensible and expressive wide-area service discovery architectures. This was accomplished through the design of a general and generic service discovery architecture named WPSD that relies on event-based communications. The applicability of WPSD to Personal Networks (PNs) was examined along with interoperability requirements with other discovery frameworks. WPSD can enable service discovery between remotely located PN clusters and can facilitate PN establishment and context information delivery. WPSD can easily interoperate with popular service discovery protocols to fulfill the requirements of wide-area service discovery in pervasive environments. These WPSD features should ease large scale deployment and adoption.

6 Acknowledgments

The authors would like to thank the reviewers for their valuable comments. This work was partially funded by IST Integrated Project MAGNET No. 507102.

7 References

- 1 My personal Adaptive Global NET. IST-MAG-NET consortium. Available from: <http://www.ist-magnet.org/>
- 2 UPnP Forum Website. <http://www.upnp.org/>
- 3 Jini Website. <http://www.jini.org>
- 4 Pietzuch, P R. *Hermes: A Scalable Event-based Middleware*. Queens' College, University of Cambridge, UK, 2004. (PhD thesis)
- 5 Fiorentino, C, Cilia, M, Fiege, L, Buchmann, A. Building a Configurable Publish/Subscribe Notification Service. *IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'05)*, Athens, Greece, June 2005. Springer-Verlag. (LNCS 3543)
- 6 Carvalho, N, Araújo, F, Rodrigues, L. Scalable QoS-Based Event Routing in Publish-Subscribe Systems. *IEEE International Symposium on Network Computing and Applications (NCA '05)*, Boston, MA, USA, 2005.
- 7 Bacon, J, Moody, K, Yao, W. A model of OASIS role-based access control and its support for active security. *ACM Transactions on Information and System Security (TISSEC)*, 5 (4), 492–540, 2002.
- 8 Belokosztolszki, A, Eyers, D M, Pietzuch, P R, Bacon, J, Moody, K. Role-based access control for publish/subscribe middleware architectures. *International Workshop on Distributed Event-Based Systems (DEBS'03)*, ACM SIGMOD, San Diego, CA, USA, June 2003. ACM.
- 9 Louati, W, Girod Genet, M, Zeghlache, D. UPnP extension for wide-area service discovery using the INS/Twine Framework. *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'05)*, Germany, September 2005.
- 10 Ghader, M, Olsen, R, Prasad, V, Jacobsson, M, Sanchez, L, Lanza, J, Louati, W, Girod-Genet, M, Zeghlache, D, Tafazolli, R. Service Discovery in Personal Networks; design, implementation and analysis. *IST Mobile Summit 2006*, Myconos, Greece, June 2006.
- 11 Meier, R, Cahill, V. Taxonomy of Distributed Event-Based Programming Systems. *The Computer Journal*, 48, 602–626, 2005.
- 12 Buford, J, Ross, K, Kolberg, M. *CORE Subgroup Problem Statement*. Internet Draft, January 2006. draft-irtf-p2prg-coreproblem-statement-00
- 13 Carzaniga, A, Rosenblum, D, Wolf, A L. Challenges for Distributed Event Services: Scalability vs. Expressiveness. *Engineering Distributed Objects (EDO '99)*, ICSE 99 Workshop, Los Angeles, CA, May 1999.
- 14 Stoica, I, Morris, R, Karger, D, Kaashoek, M F, Balakrishnan, H. Chord: A scalable peer-to-peer lookup service for internet applications. In: *Proceedings of the 2001 ACM SIGCOMM Conference*, 149–160, San Diego, CA, August 2001.
- 15 Ratnasamy, S, Francis, P, Handley, M, Karp, R, Shenker, S. A scalable content addressable network. In: *Proc. of the ACM SIGCOMM Conference*, 161–172, San Diego, CA, 2001.
- 16 Rowstron, A, Druschel, P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218, 329–350, November 2001.
- 17 Zhao, B Y, Huang, L, Stribling, J, Rhea, S C, Joseph, A D, Kubiatowicz, J D. Tapestry: A resilient globalscale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22 (1), 41–53, 2004.
- 18 Wang, C, Carzaniga, A, Evans, D, Wolf, A. Security Issues and Requirements in Internet-scale Publishsubscribe Systems. In: *Proceedings of the Thirty-Fifth Annual Hawaii International Conference on System Sciences (HICSS'02)*, 303.
- 19 Miklos, Z. Towards an access control mechanism for wide-area publish/subscribe systems. *International Workshop on Distributed Event-based Systems (DEBS'02)*, Vienna, Austria, July 2002.
- 20 Fiege, L, Zeidler, A, Buchmann, A, Kehr, R K, Mhl, G. Security Aspects in Publish/Subscribe Systems. *International Workshop on Distributed Event-Based Systems (DEBS'04)*, May 2004.
- 21 Balazinska, M, Balakrishnan, H, Karger, D. INS/Twine: A scalable peer-to-peer architecture for intentional resource discovery. *International Conference on Pervasive Computing*, 195–210, Zurich, Switzerland, August 2002. Springer-Verlag.

- 22 Robinson, R, Indulska, J. Superstring: A Scalable Service Discovery Protocol for the Wide-Area Pervasive Environment. *Proc of the 11th IEEE International Conference on Networks, ICON'03*, 699–704, Sydney, Australia.
- 23 Hodes, T D, Czerwinski, S E, Zhao, B Y, Joseph, A D, Katz, R H. An Architecture for Secure Wide-Area Service Discovery. *Wireless Networks*, 8, 213–230, March 2002.
- 24 Lee, C, Helal, A. A Multi-tier Ubiquitous Service Discovery Protocol for Mobile Clients. *Proceedings of the 2003 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2003)*, Montréal, Canada, July 2003.
- 25 Arabshian, K, Schulzrinne, H. GloServ: Global service discovery architecture. In: *MobiQuitous*, 319–325. IEEE Computer Society, June 2004.
- 26 Zhu, F, Mutka, M, Ni, L. Service Discovery in Pervasive Computing Environments. *IEEE Pervasive Computing*, vol. 4, pp. 81-90, 2005.
- 27 Louati, W et al. Networking in Personal Networks. *Workshop on Applications and Services in Wireless Networks (ASWN'05)*, Paris, France, June 29 – July 1, 2005.
- 28 Louati, W, Zeghlache, D. A Dynamic VPN management architecture for Personal Networks. *ASWN'05*, Paris, France, June 29 – July 1, 2005.
- 29 Murakami, H, Olsen, R L, Schwefel, H P, Prasad, R. Managing Personal Network Specific Addresses in Naming Schemes. *WPMC'05*, Aalborg, Denmark, September 2005.

Wassef Louati received the MS Degree in Computer Science in 2004 from Pierre et Marie Curie University (Paris, France). Currently, he is a PhD student at the "Institut National des Telecommunications" (Evry, France) in the wireless networks and multimedia services department. His main research interests include naming, service discovery and peer-to-peer architectures with a current focus on Personal Networks.

email: wassef.louati@int-evry.fr

Djamal Zeghlache graduated from SMU in Dallas, Texas in 1987 with a PhD in Electrical Engineering and the same year joined Cleveland State University as an Assistant Professor. In 1990 and 1991 he worked with the NASA Lewis Research Centre on mobile satellite terminals, systems and applications. In 1992 he joined the Networks and Services Department at INT where he currently acts as Professor and Head of the Wireless Networks and Multimedia Services Department. He is an active member of the IEEE communications Society and a member of the IEEE Technical Committee on Personal Communications. He acted as co-technical chair of the ASWN 2001, 2002, 2005 Workshops and Technical Chair of the Wireless Communications Symposium for Globecom 2003. He acts as lead scientist for INT in the European project MAGNET Beyond. He is also an expert group member of the eMobility Platform at the European level for framework program 7 and involved in WWRP working groups 2, 3 and 6. His interests and research activities span a broad spectrum of issues related to wireless networks and services. The current focus besides resource allocation is on dynamic adaptation and configuration of wireless networks and services based on context awareness and service discovery using P2P and autonomic networking paradigms. An ongoing activity relates to personal networks seen as a wide area extension of wireless personal area networks involving remotely located personal clusters. A key objective is to address the challenge of establishing overlay networks and service overlays for these networks at run time to enable dynamic and context aware adaptation of services and applications.

email: djamal.zeghlache@int-evry.fr