Remote Service Discovery and Control for Ubiquitous Service Environments in Next-Generation Networks

Andreas Häber

Remote Service Discovery and Control for Ubiquitous Service Environments in Next-Generation Networks

Doctoral Dissertation

University of Agder
Faculty of Engineering and Science
2010

Doctoral Dissertations at the University of Agder 21

ISSN: 1504-9272
ISBN: 978-82-7117-665-5

Printed by the Printing Office, Univeristy of Agder
Kristiansand

Dedication

To my families.

Acknowledgement

Although this dissertation is authored by me, I would not be able to do so without the help and support from several people that I would like to thank here.

I want to start with thanking Prof. Frank Reichert for the excellent mentoring these years. You have an incredible ability to inspire and make work fun. Also thanks for introducing me to great music bands, such as Devildriver. I feel very honoured and proud of having had you as my supervisor.

Thanks also to my co-supervisors, Prof. Andreas Prinz (University of Agder) and Dr. Andreas Fasbender (Ericsson), for all your help and support.

I would also like to thank the University of Agder and all colleagues for the help, support and collaboration. The administration is always helpful and friendly, likewise the library and other support functions. In particular, thanks to my colleagues in the Agder Mobility Lab research group.

Thanks to Ericsson for the excellent cooperation all these years! I have always felt welcomed at the research labs I have visited around the globe (Aachen, Germany; Kista, Sweden; Montreal, Canada). In particular, I want to thank Dr. Andreas Fasbender and Martin Gerdes of Ericsson Eurolab in Aachen, Germany. It has been both exciting and a pleasure to work together with you!

Thanks to my families, both in the west and in the east, for all your support!

Finally, I am very grateful for all your support and understanding, my dear 张力 (Li). Thanks for always being there for me!

Yours Truly,

Andreas Häber

Table of Contents

Publication list

This dissertation consists of six publications and an elaboration of these. The publications are referred to in the text using their Roman numerals I-VI.

[I] A. Häber, M. Gerdes, F. Reichert, R. Kumar, and A. Fasbender, "Remote Service Usage through SIP with Multimedia Access as an Use Case," 18th IEEE Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2007), Athens, Greece, 2007.

[II] A. Häber, M. Gerdes, F. Reichert, A. Fasbender, and R. Kumar, "Using SIP Presence for Remote Service Awareness," Norsk Informatikkonferanse 2008 (NIK'08), Kristiansand, Norway, 2008.

[III] A. Häber, J. R. D. M. Gómez, and F. Reichert, "Virtualization of Remote Devices and Services in Residential Networks," 3rd International Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST 2009), Cardiff, Wales, 2009.

[IV] A. Fasbender, S. Hoferer, M. Gerdes, T. Matsumura, A. Häber, and F. Reichert, "Phone-controlled Delivery of NGN Services into Residential Environments," 2nd IEEE Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST 2008), Cardiff, Wales, 2008. (Best paper award)

[V] A. Fasbender, M. Gerdes, T. Matsumura, A. Häber, and F. Reichert, "Media Delivery to Remote Renderers Controlled by the Mobile Phone," 6th IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 2009. (Demonstration)

[VI] A. Häber, M. Gerdes, F. Reichert, A. Fasbender, and R. Kumar, "Delivering Services to Residential Appliances by Utilizing Remote Resource Awareness," 2nd IEEE Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST 2008), Cardiff, Wales, 2008.

[VII] M. Gerdes, A. Fasbender, A. Häber, and F. Reichert, “A Method and Apparatus for Service Discovery”, International patent application WO/2008/082346, to Telefonaktiebolaget LM Ericsson, World Intellectual Property Organization, 2008.

Table of Figures

Table of Tables

Abbreviations

3rd Generation Partnership Project
3GPP 4

Application Programming Interface
API 15

Authentication, Authorization and Accounting
AAA 18

Connected Device Configuration
CDC 74

Connected Limited Device Configuration
CLDC 64

Consumer Electronics Association
CEA 13

Customer Premises Equipment
CPE 11

Device Control Protocols
DCP 15

Dynamic Host Configuration Protocol
DHCP 92

Ericsson IMS Client Platform
ICP 78

European Telecommunication Standardisation Institute
ETSI 17

Extensible Markup Language
XML 21

Gateway GPRS Support Node
GGSN 100

Generic Bootstrapping Architecture
GBA 93

Generic Event and Notification Architecture
GENA A-3

Global Positioning System
GPS 11

Heating, Ventilation and Air Conditioning
HVAC 29

HIGA for Remote Access
HIGA-RA 73

High-Speed Packet Access
HSPA 4

Home Gateway Initiative
HGI 10

Home IMS Gateway
HIGA 73

Hypertext Transfer Protocol
HTTP 14

IM Public User identity
IMPU 93

Integrated Services Digital Network
ISDN 15

Chapter I

Introduction

This chapter introduces the research area and defines the research questions. In addition, scientific contributions and an outline of the thesis are given.

1.1 Positioning the research

Technology has always shaped society. And the development of transistors and microprocessors that started the age of digital technology will change it even more.

Consumer appliances for entertainment and communication are becoming digital. Analog cameras with films are replaced by digital cameras using storage chips. Digitalization has opened up new possibilities by interconnecting appliances and simplifying content handling. Photo albums and CD collections are being replaced by media servers. “eBook-readers” contain thousands of books. TVs and digital photo frames display photos from local sources and remote Internet services. Users benefit significantly from their appliances being interconnected and cooperating with each other.

At the same time homes are becoming more intelligent thanks to new home automation [1] technology. This technology supports deploying

Fig. I-1 Digitalization of home services.

sensors and actuators in buildings, for example to control heating, ventilation and lighting.

Thus, ubiquitous computing [2, 3], which is the distribution of computing systems in environments that appear seamless to users, is already appearing in homes. The concept of ubiquitous service environments [4] extends ubiquitous computing to services in logical and physical spaces.

Both residential appliances and sensors are devices that provide services. By adding networking, these devices can cooperate and enable more complex services. Before services can cooperate, they need to be aware of each other. Static configuration is possible, but does not scale well and is difficult to manage.

Service discovery mechanisms allow relationships to be handled dynamically. In general, service discovery protocols let services announce themselves and let services be discovered by controllers that listen to these announcements. Example service discovery mechanisms are included in: Universal Plug & Play (UPnP) [5-7], Bluetooth [8-10], Service Location Protocol (SLP) [11], ZigBee and Zero Configuration Networking (ZeroConfig) [12].

Mobile communication devices, like mobile phones have undergone a tremendous development. From being a niche appliance, mobile phones have become ubiquitous appliances that are used by millions of people all over the world. Studies show that mobile phones are now used as a fashion statement [13] and people feel lost without their mobile phone nearby [14].

Mobile phones are constantly being improved to become smaller,

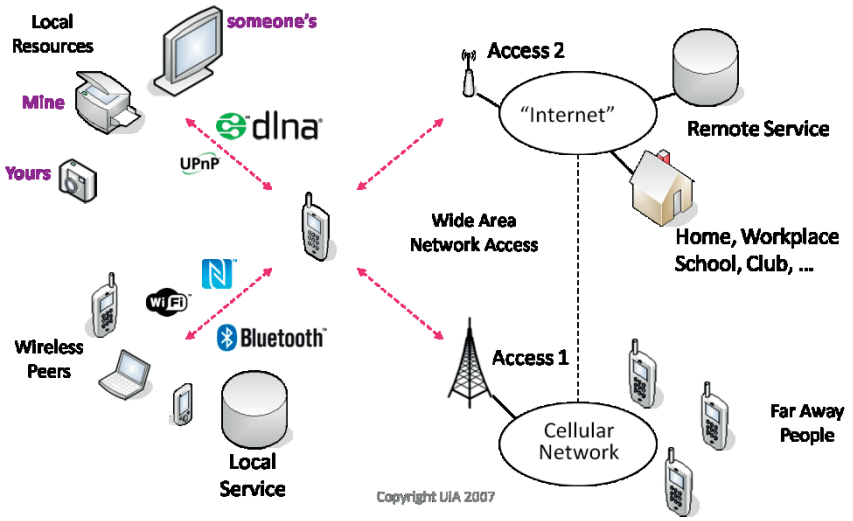


Fig. I-2 Mobile phone allows both bridging distances and interacting with the immediate environment.

faster and increasingly feature rich. Commonly mobile phones have been used to bridge distances, to contact far away people and services, as shown at the right-hand side of Fig. I-2. Because of the ubiquitous nature of mobile phones, they are ideal devices to interact with the local environment, which is shown at the left-hand side of Fig. I-2.

Wireless communication systems are constantly improving as well. The 3rd Generation Partnership Project (3GPP) is responsible for developing several of these, including the Universal Mobile Telecommunications System (UMTS) and the High-Speed Packet Access (HSPA). The latter offers theoretical peak data rates of 14.4 Mbps in the downlink and 5.76 Mbps in the uplink. Recently 3GPP have released their first fourth generation radio access technology, Long Term Evolution (LTE) [15]. LTE has been designed to provide theoretical peak data rates of 300 Mbps in the downlink and 75 Mbps in the uplink, as well as better support for moving terminals. The increased bandwidth and low latency offered by these technologies provide for a better user experience, and allow users to stay connected anywhere at anytime.

A key principle in converged fixed-mobile "Next-Generation Networks" (NGN) is the separation of data transport and service control, as shown in Fig. I-3 (based on Figure 1 of [16]). At the heart of service control lies the IP Multimedia Subsystem (IMS) [17, 18] as defined by 3GPP. IMS uses the Session Initiation Protocol (SIP) [19], developed by the Internet Engineering Task Force (IETF), as the core signaling protocol to provide control and integration of different services. In addition,

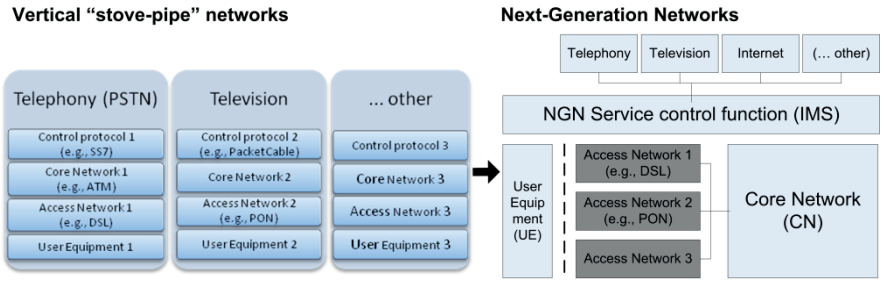


Fig. I-3 Migration to NGN and separation of transport and services in NGN.

IMS provides access control and Quality of Service (QoS) for services and supports gateways to interact with other systems.

Users should ideally be able to access their residential services from anywhere using any type of terminal. They will benefit from capability signaling offered by NGN as service providers can now adapt service delivery to different types of appliances and communication links and thereby provide a better "quality of experience".

Today solutions for remote service discovery and control do not integrate features of ubiquitous computing and next generation networks. Without these basic key functions users will not be able to use their services and digital media with local appliances in their homes, at their friends' home, or wherever they are.

1.2 Research question

The main research question of this dissertation is:

> ***"How can ubiquitous computing and next-generation networks benefit from each other?"***

The basic assumption behind this question is that an integration of both areas will create new business opportunities and end-user

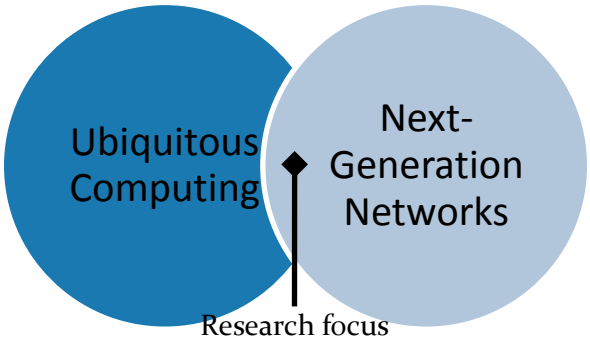


Fig. I-4 The benefits in how ubiquitous computing overlaps with NGN is the research question this dissertation tries to answer.

benefits. This dissertation tries to prove that taking advantage of components and mechanisms from both areas will produce efficient scalable solutions. Other core architectural qualitities, like scalability, migration, security, trust and reliability also have to be addressed.

A visual representation of the main research question is given in Fig. I-4.

1.3 Objective and scope of the research

The main objective of this research is to investigate how ubiquitous service environments could be connected together with support from NGN.

NGN-aspects have been limited to service layer aspects.

Use cases and feasibility studies have to a large extent been consumer-oriented.

1.4 Scientific Contributions

The contributions made within this thesis are divided into two categories. Firstly, use cases, architectures and protocols for remote service discovery and control have been developed. The solution integrates mechanisms from ubiquitous computing with enablers offered by Next-Generation Networks. These are covered mainly by the publications [I], [II], [III] and [VII].

The other contribution category focuses on the role of service providers in publications [IV], [V] and [VI]. [IV] and [V] describe a solution that separates content delivery from session control. [VI] presents solution design alternatives to let service providers take advantage of residential services and capabilities.

1.5 Thesis Outline

In Chapter II technologies and trends related to the research are described. Next, in Chapter III use cases that motivate remote service discovery and control are shown. Requirements for solutions are covered in the second part of that chapter. In Chapter IV, a solution for remote service discovery and control is described along with an evaluation and compared with an alternative solution. Chapter V covers the objective, implementation and findings of several feasibility studies that were undertaken to understand use cases and business roles for remote service discovery, in addition to validate the developed protocols. Finally, Chapter VI concludes the dissertation.

Chapter II

Technologies and trends

Covers essential trends and technologies related to the thesis topics. Topics related to residential networks are covered first, as an example environment of ubiquitous computing. Next, service discovery technologies are described. Thereafter Next-Generation Networks are covered with an emphasis on the service layer.

2.1 Residential networks and gateways

When the personal computer was introduced in 1981 only few households owned a single computer. As the Internet became commonplace, with attractive services and content available, intended not only for researchers and enterprises, but also targeting consumers, it became more important for households to acquire a personal computer as well. The trend today, especially in developed countries, is [20] that households are moving away from sharing a single personal computer to a network of computers and other appliances, which is known as a residential network.

An example residential network is given in Fig. II-1. Residential gateways (RGw) [21, 22] connect Local Area Networks (LAN) to Wide Area Networks (WAN) that are offered by access network operators. They range from simple devices, like broadband modems, to sophisticated devices that take care of network security, advanced routing and private network management.

In addition to providing Internet connectivity, residential gateways are becoming a generic service platform [21, 23-25] for home networks. This migration is motivated by the requirements of several residential services that require certain new functionality in the home, such as IP telephony and IPTV. Instead of one dedicated appliance for each such service, the residential gateway can handle all of them at the same time. An important forum in this area is the Home Gateway Initiative (HGI) [26], which was launched by the telecommunication industry for joint-development of specifications for residential gateways. Due to the

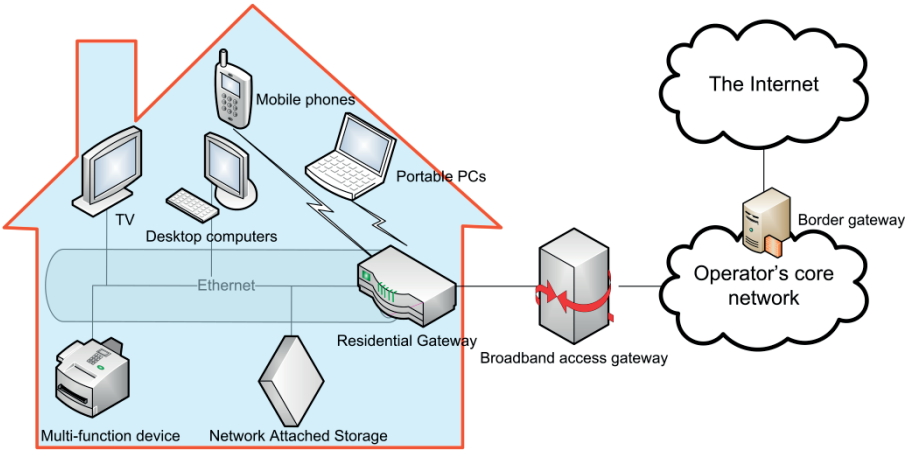


Fig. II-1 Example of a residential network. Devices are connected to a residential gateway that inter-connects them to the operator that provides Internet access.

central role of residential gateways, remote diagnostic and management functionality is increasingly being included to allow service providers to support their customers remotely. A broadly adopted solution is the Broadband Forum's Customer Premises Equipment (CPE) WAN Management Protocol [27-29] that allows operators to monitor their customer's networks, diagnose and fix issues remotely.

2.2 Service discovery

In general, service discovery deals with locating and identiyfying services that can be used to accomplish a task. For example, discovering a printer to print documents on, or locating a shop offering the service of hairdressing. When a service has been located, a reference is usually obtained (i.e., an address) that can be used to communicate with the service offered and request more service description details.

From the perspective of "daily human life", service discovery is usually achieved by querying a directory service, looking at service announcements, or asking other people. Traditionally, the first approach is achieved by looking up in the "yellow pages" or calling a service discovery company, like the Norwegian "1881" service. In addition, Global Positioning System (GPS) devices and Internet map search sites, such as Google Maps [30], allow users to perform similar searches themselves. The second approach is similar to posters and road signs that describe services and give directions to locate them.

Multicast searching is commonly used in dispatch systems, such as used by rescue teams, taxi companys and police.

Similar mechanisms have been developed for computer networks. In [31], 26 service discovery protocols are compared, and several more do exist. To evaluate and compare these protocols F. Zhu et al. [32] has developed a taxonomy based on the existing protocols, which is reproduced in Fig. II-2.

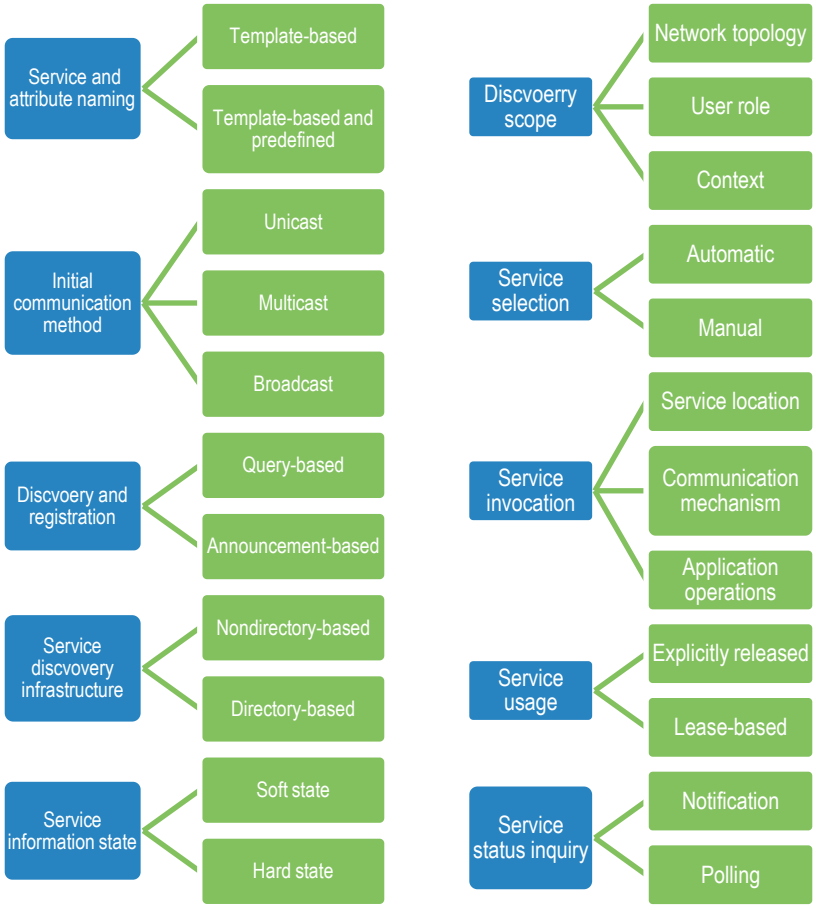


Fig. II-2 Taxonomy of service discovery protocol components.

2.2.1 Service discovery and consumer appliances

Most notable of the service discovery protocols used in the consumer space are Bluetooth (at least 4355 products [33]) and UPnP (more than 5500 products have been approved by DLNA [34]). In addition, Apple's Bonjour is increasingly becoming popular as well, fueld by other popular Apple products, such as iTunes together with the iPod and the iPhone.

In the following section, UPnP is covered as an example due its importance in the market. For the same reason UPnP has been used in the prototypes covered in [I-VI] as well.

2.2.2 Universal Plug & Play

Universal Plug & Play (UPnP) is used both to refer to a technology, the UPnP Device Architecture (UDA) [5], and an organization, the UPnP Forum. Since the UPnP Forum was established in 1999, the technology is increasingly used in various appliances, especially consumer appliances such as residential gateways and multimedia products. As of 31. July 2009 there are 884 forum members. UPnP technology is used by many organizations, such as DLNA, the Broadband Forum, CableLabs, and the Consumer Electronics Association (CEA). In September 2007, the UPnP Device Architecture, and other standards by the UPnP Forum, were approved by the International Standards Organization (ISO) / International Electrotechnical Commission (IEC) Joint Technical Committee (JTC) 1 (ISO/IEC JTC 1) as a Publicly Available Specification (PAS) [35]. This is important because it shows that UPnP technology is widely used and since it allows other

international standardization bodies, such as the International Telecommunication Union (ITU), to refer to it in their standards.

Logical model

UPnP is based on devices that are discovered and controlled by control points. As shown in Fig. II-3, devices are resources that may contain services and embed other devices. The top-level device is called a root-device. For example, consider a TV. It can support both to serve the media received from its tuner and render media received from other sources. Following UPnP, these are two different logical devices and may be embedded inside the root device, which is the TV.

Architecture

UPnP is leveraging web technologies, such as the Transmission Control Protocol (TCP) [36] / the Internet Protocol (IP) [37] (TCP/IP), the Hypertext Transfer Protocol (HTTP) [38] and the Simple Object Access Protocol (SOAP)[1] [39]. The UPnP Architecture specifies several

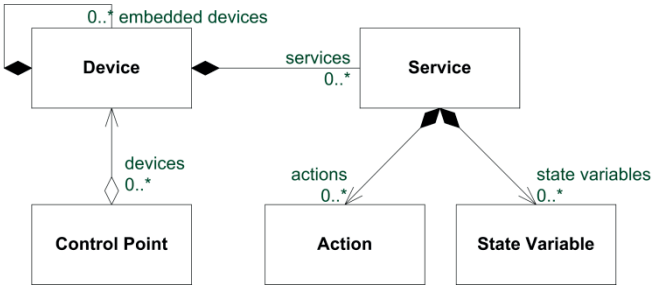


Fig. II-3 Logical model of UPnP.

[1] Since version 1.2 SOAP is the name of this protocol instead of an acronym for "Simple Object Access Protocol".

extensions to these protocols, in particular HTTP. These extensions are introduced below when describing the UPnP Device Architecture. Layered on top of the device architecture is Device Control Protocols (DCP) that are standardized by the UPnP Forum and described in more detail in 6.3Appendix A. Vendor-specific extensions may be added to these DCPs for extra functionality, such as adding a proprietary search control action to a service. Thus, devices can interoperate with other vendor's equipment, but will work best together with control poitns aware of these extensions.

Because the UPnP Forum only specifies and standardizes protocols, it is up to vendors to provide application programming interfaces (API) to take advantage of it. Several such APIs are available for different platforms [40].

2.3 Next-Generation Networks

Traditionally, services have been thightly coupled with a specific transport network and signaling protocol [41-43], such as Integrated Services Digital Network (ISDN) [44] and terrestrial television broadcasting. However, service delivery over the Internet has tremendously changed the playing field. Reasons for this change include that the Internet is designed to be independent of underlying network. In addition, the pervasiveness of the Internet permits services to be delivered to any terminal anywhere. A core principle of the Internet is to keep intelligence at the edges of the network. Contrary to most traditional service solutions, like telephony and cable television, anyone can be a service provider in the Internet. Therefore, traditional

service and network providers have been left with the role of providing users with network access to the Internet. This role is undesirable for these players because their customers pay for services, and therefore their revenue is reduced when customers use cheaper services offered over the Internet.

On the other hand, the Internet has got short-comings of its own [45, 46]. For instance, the Internet does not enforce end-to-end QoS. Also there is currently no solution for the Internet that addresses all mobility requirements [47]. Moreover, there is a lack of a common trust, privacy and security approach.

To remedy these challenging issues the "Next-Generation Network (NGN)" has been developed. The International Telecommunication Union (ITU)-Telecommunication Standardization Sector (ITU-T) is leading the standardization of NGN together with regional standardization bodies and related fora. The first recommendations on NGN by ITU-T were approved in late 2004. Release 1 was completed in late 2005. Recommendation Y.2001 "General of NGN" [48] gives the following definition of NGN:

> *"A packet-based network able to provide telecommunication services and able to make use of multiple broadband, QoS-enabled transport technologies, and in which service-related functions are independent from underlying transport-related technologies. It enables unfettered access for users to networks and to competing service providers and/or services of their choice. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users."*

A core characteristic of NGN is the indepency between the transport and service areas, which are named the transport stratum and the service stratum, and illustrated in Fig. II-4. This separation gives the necessary decoupling between services and transport networks that is lacking in traditional networks.

In the following, the service control function of the service stratum is covered in more depth due to its relevance to the rest of this dissertation.

2.3.1 IP Multimedia Subsystem: the service control function

IMS [17] was originally developed by 3GPP as part of the third generation (3G) core network to merge cellular networks with the Internet. Since then, it has also been endorsed by several Standardization Development Organizations (SDO). Examples include the Open Mobile Alliance's (OMA) OMA IP Multimedia Subsystem (IMS in OMA) [49], CableLabs' PacketCable 2.0 [50] and the European Telecommunication Standardisation Institute's (ETSI) Telecoms & Internet converged Services & Protocols for Advanced Network (TISPAN) [51]. Moreover, it has also been adopted by ITU-T [52] as the

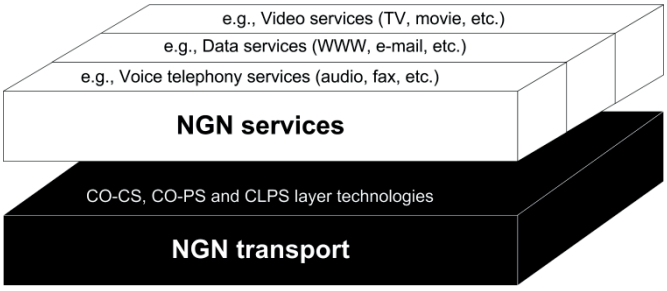


Fig. II-4 Separation of services from transport in NGN. Based on Figure 1 of [16].

NGN service control function.

As 3GPP seeks to converge Internet and telecommunication services, it has adopted protocols developed by the Internet Engineering Task Force (IETF) and setup a strong collaboration [53]. This collaboration aids to ensure interoperability with the Internet. An example of this collaboration has been continuous development of the Session Initiation Protocol [19], which was selected by the 3GPP as the core signaling protocol of IMS Thereby, IMS can be seen as a SIP-network that overlays the access network functionality. Another core protocol is the Diameter Base Protocol [54], which is used for authentication, authorization and accounting (AAA).

IMS supports service deployment by both network operators and third party service providers. Thus, a variety of services can be developed independently and at the same time utilize the common features of the IMS infrastructure. With the application triggering architecture [55] integrated services are supported in IMS.

The IMS architecture is described in more detail in Appendix B.

2.3.2 NGN Presence service

The presence service is here introduced both as an example of an IMS service and because it is a core enabler for the remote service discovery solution that is given in Chapter IV. In the following, basic terms and concepts of the presence service are introduced. Furthermore, core signalling and control protocols are described.

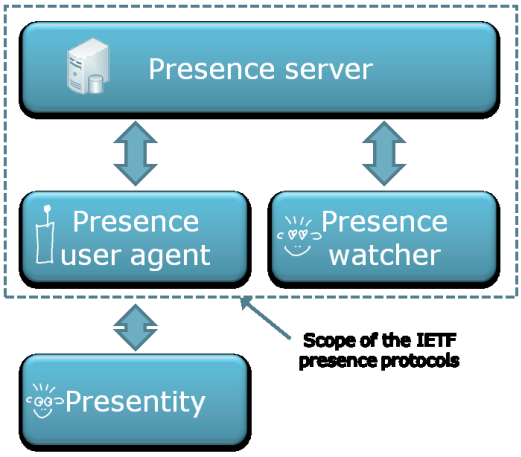


Fig. II-5 Logical entities of the presence service.

These mechanisms and technologies are important for ubiquitous service environments, where users and services may come and go at any time.

Presence service functionality

The presence service allows entities to receive notifications on presence state updates sent by other entities. One example is to receive a presence state update that a friend's phone is offline. This state change indicates that users do not need to try to dial a friend's phone, and use non-instant communication mechanisms, such as e-mail, instead.

As shown in Fig. II-5, there are four entities involved in the presence service architecture. The presence entity (presentity) provides presence information to a presence service. A presentity can be anything that has presence state, including people, buildings and communities. Furthermore, presentities can have numerous presence user agents that send state changes to the service. For example, persons can use

both a mobile phone and a netbook as their presence user agents. The presence server is responsible to collect all these state changes together and compose a combined view. All presence watchers are notified by the presence server when the presence state changes. Finally, notice that there is no protocol defined for the relation between presence user agents and presentities. This opens up for a wide range of possibilities for how presence user agents are aware of presentities and their presence state.

In addition, presence watchers can set subscription filters [56, 57] to restrict the set of presence information they are interested in. Such filters can significantly reduce traffic for sending state information updates to presence watchers. These filters are specified as queries in the W3C XQuery language [58]. Moreover, users can set policies to control access to presence information for its presentities. Thus, the set of presence information presence watchers obtain is restricted by both the access policy and the subscription filter.

In general, the SIP Event framework is used to communicate presence updates. This framework extends SIP with mechanisms that allow user agents to publish, subscribe and notify state change. Presence watchers subscribe to the presence service to receive notifications. Presence user agents publish these state changes to the presence server.

The SIP Event framework sets a lifetime to event states. Therefore, event states have to be kept alive by regularly refreshing them or by updating them; otherwise they will be deleted. Although presence state changes are mostly dynamic, it is also necessary to set presence state static. For example, some presence state almost never changes, such as

e-mail addresses at companies and Short Message Service (SMS) [59]. Another example is to provision the default state for presentities. These use cases are all enabled by a separate control protocol, the Extensible Markup Language (XML) Configuration Access Protocol (XCAP). Presence state is encoded as an XML document and XCAP let entities manipulate these documents. This protocol is based on HTTP and allows retrieving parts of an XML document (GET method), add a new part to an XML document (PUT method) and remove part of an XML document (DELETE method).

To summarize, SIP and XCAP together allows a user to publish temporal and permanent presence state for presentities to a presence service. The presence watchers receive notificatsion from the presence service when the presence state changes, restricted by any filters set.

Presence information

State changes for presentities are encoded as an XML document in a format named Presence Information Data Format (PIDF) [60]. PIDF itself is a base format that includes a minimal set of presence status information. This set includes a presentity identifier, availability status (open or closed) and optionally communication addresses (e.g., phone numbers and e-mail addresses). PIDF defines several extensibility points to add information to a document. Examples of such extensions include the Rich Presence Extensions to PIDF (RPIDF) [61] that adds optional elements for presentities and GEOPRIV [62] for adding geographic information.

Chapter III

Use cases and solution requirements

Use cases developed for remote service discovery and control are described along with high-level requirements for functionality and quality.

3.1 Use cases

In the following sections, we describe use cases that motivate remote service discovery and control. An emphasis is put on use cases that include elements of ubiquitous computing environments supported by NGN.

Users move between different environments, such as starting from home via a commute to the office, or from a hotel to a conference center. Some environments offer few services, wheras others offer a broad range of services, as shown in Fig. III-1. Mobile terminals must be aware of services offered in the current environment to let users utilize them. Furthermore, to enable services providers that follow NGN standards to deliver directly into ubiquitous environments they need to be aware of the capabilities of that environment. Exporting capabilities of ubiquitous service environments to NGN facilitates these use cases.

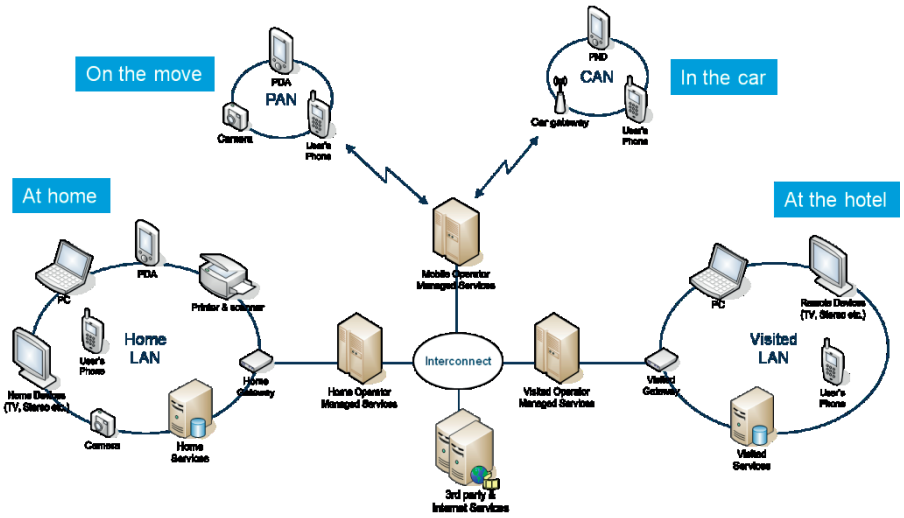


Fig. III-1 Services delivered to any network and device, and controlled by users' terminal.

3.1.1 Remote multimedia access

DLNA, along with other players, has enabled multimedia use cases that combine media players with media servers in a local domain, optionally with a separate control device. These are known as the 2-box and 3-box models respectively. Moving one or more of these entities outside of the local domain enables several new multimedia scenarios.

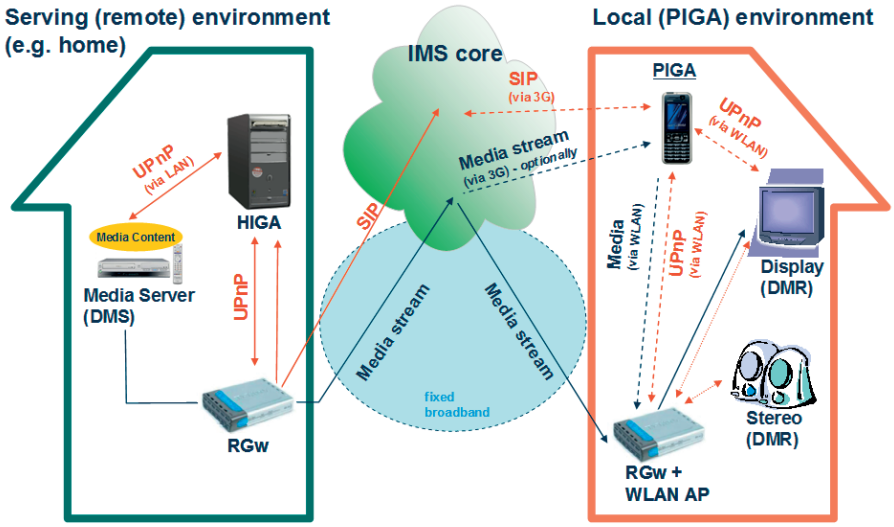


Fig. III-2 Remote media access use case, from paper [I]. Two gateways, the Home IMS Gateway (HIGA) at home and the Portable IMS Gateway (PIGA) in the local environment, facilitate remote service discovery supported by the core network. Thereby content from the home media server can be streamed to the stereo in the local environment.

Consider Alice visiting her good friend Bob. During the visit, she would like to show Bob pictures from her vaccation, which are stored at her home media server. As depicted in Fig. III-2, Alice uses her WLAN-enabled mobile phone (right hand side of the figure) to discover Bob's TV (i.e., a media player) and be aware of her home media server (left hand side of the figure) at the same time. Thereby she can browse her home media server and select her vaccation photos to be shown on Bob's TV. The media can be streamed via Bob's broadband connection, if available, or over cellular link via Alice's mobile phone.

If Alice likes some photos Bob shows her during the visit she could ask him to copy them to her media server that is now available in the domain. Doing this avoids the hassle of copying the media to an

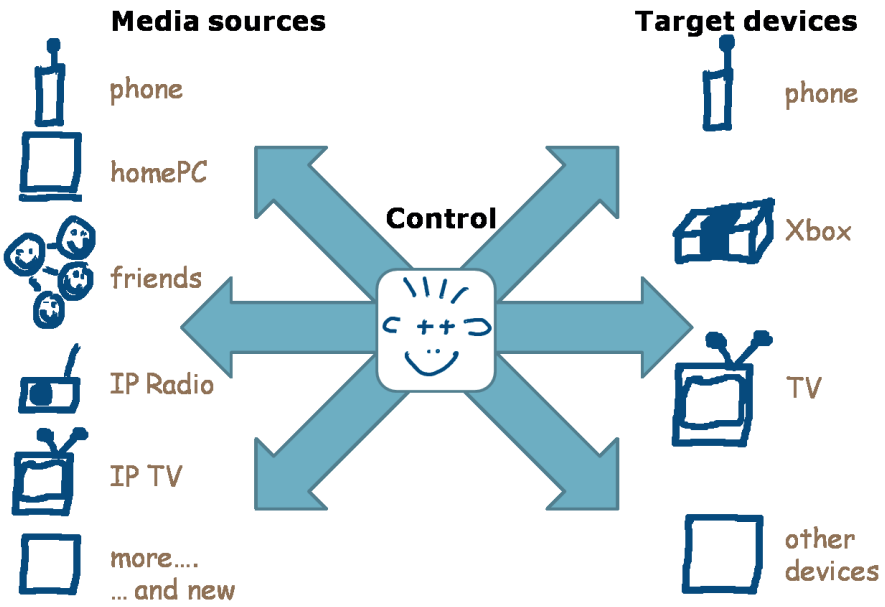


Fig. III-3 Media portal that allows users to mix media sources and target devices, based on the use case outlined in [63].

intermediate storage (e.g., her mobile phone or a USB memory stick) that Alice then would have to use to upload them to her server when she is back home.

A natural progression to these remote multimedia access use cases is to allow users to mix and match between all media sources and target devices, as shown in Fig. III-3. Such a media portal enables users full control over where media should be rendered, and possibly allow full and partial session transfer [64] among these. Such a control application is further envisioned in [63].

3.1.2 Inviting service providers home

Remote service discovery and control is not limited to services provided and consumed by consumers. For example, media content

can also be provided by media portals, as shown in paper [IV] and [V]. Similarly, professional caretakers for handling sensor information for facility management and security, and health could be involved as well. Services could be provided that act solely on the information of available services within a domain. An example is to allow device manufacturers to inform all of their customers about firmware updates. Another example is service providers that inform users that their media player is capable of rendering the content at a higher quality and offering them to upgrade, possibly for an extra charge and temporary upgrading their bandwidth limit.

Both of these examples do not include controlling the services, and accessing a subset of service information (e.g., version number and model information). Therefore, policies that grant service providers access to only the information necessary for them to provide the service is required. Policy enforcement will therefore help in restricting service providers to abuse the service information for unsolicited advertising.

3.1.3 Delivering services to an ad-hoc environment

Mobile phones that support short-range and long-range communication can enable media players in an ad-hoc environment to receive content from external sources. In the following a car scenario is considered, like the "in car" part of Fig. III-1 (at the top, right hand side of the figure).

The mobile phone facilitates accessing NGN services through users' NGN identity and related profiles. However, in-car appliances could provide better capabilities, such as screen resolution, to offer users a higher quality of experience for multimedia related services. For NGN media service providers to deliver content to these in-car appliances they have to be aware of that environment. This scenario is depicted in Fig. III-4, where the mobile phone controls the session and the media

Fig. III-4 Connected car use case.

is delivered to in-car appliances.

3.1.4 Remote facility management

It is foreseen that the typical equipment in private buildings are being connected to digital home networks to support automation and management, it becomes necessary to provide easy interfaces to control this automation support. Automation and management support includes for example, the control of Heating, Ventilation and Air Conditioning (HVAC) equipment, lightning control and digital security cameras. For instance, the UPnP Forum has standardized services to host and control such devices. This enables local control of the building and its equipment respectively.

Machine-to-machine (M2M) technologies are emerging to remotely control, configure and monitor facilities [65]. Remote service discovery and control could be an important enabler for such technologies.

Residential scenarios in a remote context include preparing buildings

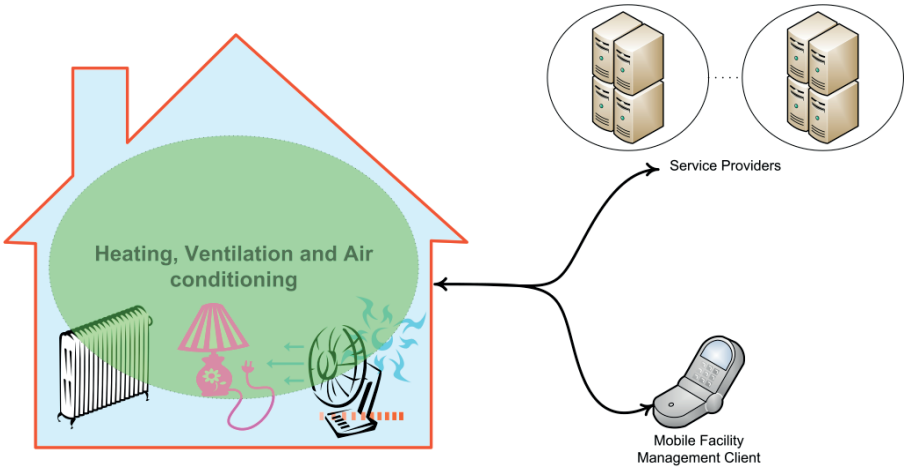


Fig. III-5 Remote facility management use case.

for arrival, such as on the way home from the office or on the way to a cabin. Travel agencies could grant their customers access to remotely manage vaccation homes for the duration of their stay through their NGN user identity.

3.2 Functional requirements

Ubiquitous service environments shall be able to discover and be aware of each other's capabilities. Similarly, NGN services shall also be able to be aware of services in ubiquitous service environments.

In addition it shall be possible to control remote services that have been discovered.

3.3 Quality requirements

In the following quality requirements [66] are given that solutions must fulfill.

3.3.1 Compatibility

[67] defines compatibility as: *"The ability of two or more systems or components to perform their required functions while sharing the same hardware or software environment."*

In this context, this quality indicates whether existing systems and interfaces in two domains will be compatible with the solution. One scenario is that a UPnP service can be controlled remotely without any changes to the UPnP service itself.

Generally, compatibility with existing systems can be achieved by adhering to current interface specifications and not introducing new

interfaces. Several service discovery mechanisms for ubiquitous computing have already been standardized, and there is already a vast amount of products on the market that implements these. Similarly, core protocols for NGN have been selected as well and operators are deploying systems based on these. Compliance with these existing interfaces adds new value to the existing systems. Furthermore, standardization becomes easier and eases the introduction of the new functionality.

3.3.2 Security

Many residential services have been designed with the constraint that they will be accessed from within the residential network. Typically it is assumed that all entities in a residential network are friendly. Therefore, these services have rather relaxed security policies.

Due to the relaxed inbuilt security capabilities it may not always be explicit who the owner of a service is. For example, the current UPnP Device Architecture does not include any authentication of users. In such cases service ownership can be inferred to the owner of the domain it resides within, such as the network administrator in the UPnP case.

However, solutions shall not only facilitate access to services for the service owner but also external sources, like friends, contacts and service providers. Therefore, access to services and information about them must be protected from unauthorized access.

3.3.3 Integrity

Users of service information will encounter problems if they are not given an accurate view of services. For example, that they are led to believe that a friend's media content service is available when in reality it has been turned off. Service providers, such as the media provider in the use case described in 3.1.2, need reliable information to provide high quality services.

Mobile user terminals move between different environments. Some environments are known, such as at home, in the car and at the office, but often users roam into unknown environments, such as a weekend at a hotel room. Service discovery protocols allow user terminals to be aware of their current environment. Therefore remote service discovery solutions must support that the environment is changing as well.

3.3.4 Scalability

High availability is important for both users and service providers. The availability and performance influence users' acceptance and quality of experience. Service providers need high availability as part of service layer agreements regarding the services they provide.

As with security, residential services are often not designed to scale up and support high demand. Therefore, solutions must ensure that the demand can be handled without overloading the services.

3.4 Summary

We have given use cases and key requirements for integration of ubiquitous service environments and NGN. These must be taken into account when designing new solutions.

Chapter IV

A new concept to enable remote service discovery and control: Service presence

This chapter introduces the service presence concept and gives an architecture for remote service discovery and control based on this new concept. Moreover, it is shown how non-remote capable control points can discover and control remote services.

4.1 Solution approach

An architecture that allows users to discover and control ubiquitous services in the local domain and in remote domains is described in this chapter. The architecture permits service providers to discover services in users' domains and adapt services to users' environment before they are delivered.

Exposing domains to the outside increases the possibility for privacy intrusion and compromise security. Therefore, trusted caretakers are needed to protect the ubiquitous service environments by enforcing access policies.

Several new functions are introduced in this architecture in addition to the standard service discovery functions services and service controllers. One new function is a distinct contact point for communicating with services within a domain, named "service

discovery gateway". Another new function makes remote services compatible with legacy service controllers by creating local, virtual representations of remote services, named "service virtualizer".

To support ubiquitous service environments with scalability and access control in remote service discovery and control, a thrusted third party provides a presence server and a presence access policy server.

4.2 Services as presentities

Traditionally [68-71] the presence service has only been considered as an enabler for dynamic address books (e.g., the OMA Converged Address Book enabler [72]), instant messaging and related communication services. However, there are no restrictions within the IETF Presence framework that presentities should be people. Therefore, this framework can be used for services as presentities as well. As shown in Fig. II-5 protocols for communication between presentities and presence user agents are outside the scope of IETF. Many mechanisms are used to infer the presence of people, such as idle time on a computer, information from calendar applications or directly configured by the people themselves.

The presence service fulfills several of the solution quality requirements making it an attractive enabler to be used to realize remote service discovery.

There is much communality between service discovery protocols and presence. Service discovery can be seen as presence awareness of services, and include ways of querying and announcing the presence of services. Following this, services can be considered as presentities like people. For people presentities typically ad-hoc protocols like a presence user agent monitors users' idle time to decide the presence status. With services, presence user agents use the more formal service discovery protocols to obtain presence information. Services' presence profile include service description information as well, which is analog to how people's presence information include contact details like e-mail addresses and phone numbers. These concepts are illustrated in Fig. IV-1, where a media player is shown as a presentity and a streaming media server is the presence watcher.

For example, UPnP devices are presentities and presence user agents include the control point function to discover the presentities. The presence information encompasses information obtained in both the

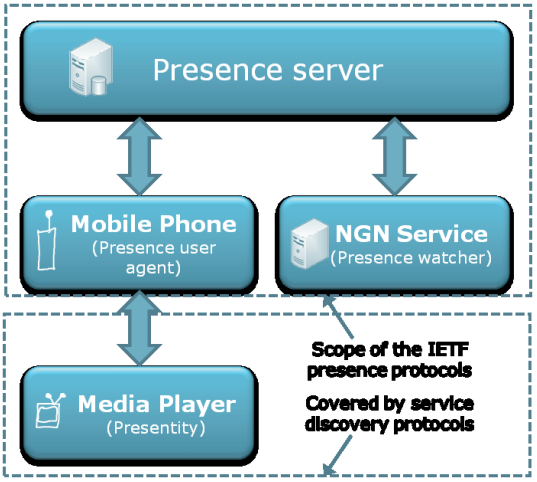


Fig. IV-1 Presence framework used for service presence.

discovery and the description phases.

4.3 System description

The logical architecture for remote service discovery and control is given in Fig. IV-2. A domain includes services and service controllers that can control these. However, service controllers are unable to directly discover and control services outside the domain. Therefore, a new function, named *service discovery gateway*, is introduced as an extended service controller that can facilitiate communication with remote domains.

An approach to develop a remote service discovery protocol could be that the service discovery gateways repeat service discovery messages

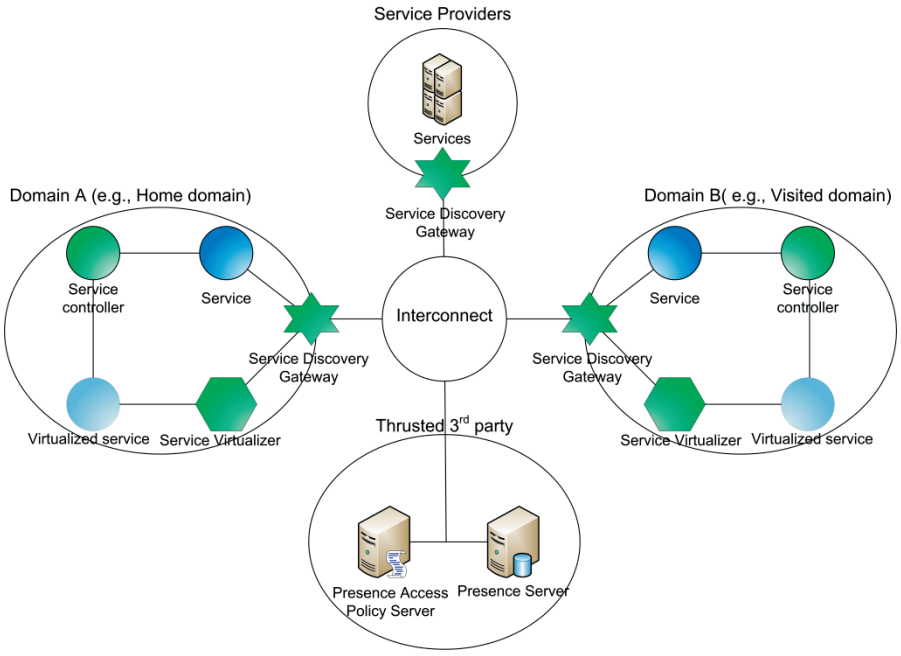


Fig. IV-2 Logical view of the remote service discovery and control architecture.

to all interconnected service discovery gateways in other domains. Consider three domains, for instance Alice's home domain, car and her friend Bob's domain. All service discovery announcements in each domain would be transmitted to the other two domains, even if no remote service controller were interested in these. Therefore, this approach is inefficient and it becomes difficult to enforce security policies on the exchanged information. Especially service discovery gateways deployed at portable devices are severly affected by this approach because precious energy resources would be consumed to transmit and receive these messages.

Since services are considered as presentities, the presence service can instead be used to exchange service information between service discovery gateways via the presence server.

The remote service discovery protocol does not include any support for legacy service controllers such as UPnP control points to be aware of the discovered remote services. This means that existing control applications have to be upgraded with support for remote service discovery, including a SIP/IMS stack. That is rather unfortunate for both consumers and vendors.

Service virtualization is a mechanism that makes remote services appear as if they were local services within a domain. A new function, named service virtualizer, is introduced that receives service presence information from a service discovery gateway and uses it to create a local, virtual representation of the remote services.

In the following, descriptions of the functional entities are given (chapter 4.4). Then the remote service discovery protocol is described (chapter 4.5). Next, access control for service presence information is covered (chapter 4.6). Using the remote service discovery protocol and access control, a remote service control protocol is described (chapter 4.7). How the concept of service virtualization works is shown thereafter (chapter 4.8). Following, a few examples for deployment of the service discovery gateway function is given (chapter 4.9). Finally, a solution alternative is described (chapter 4.10) and then these solutions are evaluated and compared (chapter 4.11) based on the requirements given in Chapter III.

4.4 Functional description

Services offer functionality that can be controlled by *service controllers* in the domain. These functions are typically found in service discovery protocols. Example usages are multimedia control applications that control media players and media servers, and smart house applications that control fans, air-conditioners and related services.

Main tasks for the service discovery gateway function are (a) be a presence user agent that publishes presence information to the presence server, (b) be a presence watcher that subscribes with the presence server to receive presence information updates, and (c) handle remote service control sessions.

The *presence server* is common to all domains and relieves the service discovery gateways from sending presence information updates to all subscribed service discovery gateways. As described below, an XCAP

[73] server for presence [74] can be connected to the presence server to support static presence information and other presence information manipulation by service discovery gateways.

Requests for service presence information are granted or denied by the *presence access policy server* by applying policy rules, for example from the operator and the user.

4.5 Remote service discovery

Service discovery gateways use service discovery protocols to collect information about local services. Typically, many services will stay in a residential network for a long time, such as TVs, picture frames and smart house infrastructure. Occassionally these services change state, such as being turned on or off, but they will still belong to the residential network. Another example is services available in a hotel room. In the following, these services are referred to as *permanent services* below. Other services that from time to time appear within a domain, such as friends visiting or guests in a hotel room, are referred to as *temporary services*.

An example of the remote service discovery protocol is given in Fig. IV-3, and discussed in the following. For simplicity the presence server and XCAP server functions are co-located in this figure, and referred to as only presence server.

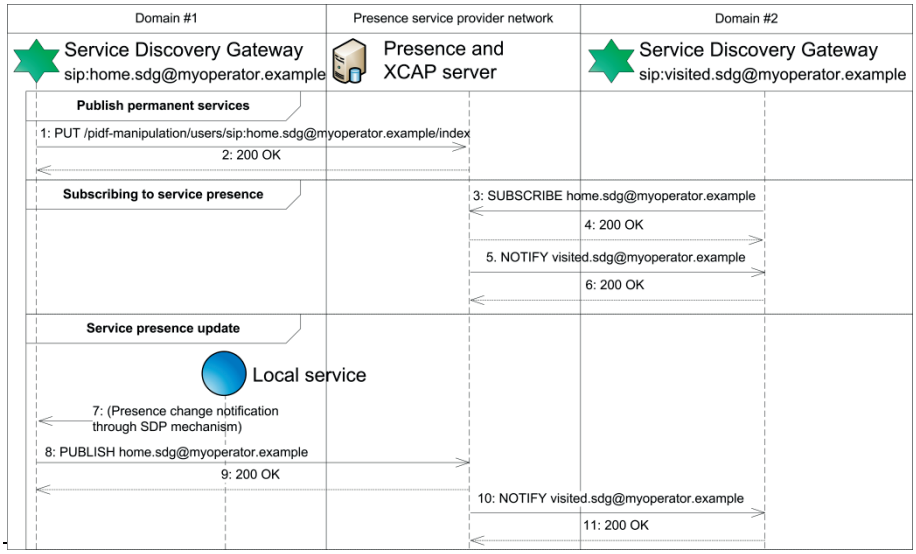


Fig. IV-3 Remote service discovery sequence.

4.5.1 Publishing service presence information

Presence information for permanent services can be published using XCAP to avoid refreshing the information. The first sequence given in Fig. IV-3, “Publish permanent services”, shows how a service discovery gateway sends the service presence information document to the presence server. Using the SIP PUBLISH method [75] temporary services and transient state changes are transmitted to the presence server, as shown in the third sequence in Fig. IV-3. This example sequence shows a local service that announces itself and subsequently the service discovery gateway publishes information about the service to the presence server.

To increase the integrity of presence information the default presence status for permanent services should be set to ‘off’ (i.e., unavailable) using XCAP. If service discovery gateways become unavailable, they

cannot refresh the lifetime for their services' current presence status. Eventually, the lifetime expires and the presence server will "fall back" to the default status set to 'off'. Although services themselves may not be offline, from external entities' point of view they should be observed as offline. Otherwise, services would be seen as online even if their presence state have changed, because in this situation the service discovery gateway does not signal such changes. Furthermore, remote service control requests cannot be handled without the service discovery gateway function available. Setting the default presence status to 'off" decrease the chance that such requests will be sent when service discovery gateways are unavailable.

To reduce the amount of information sent for service presence changes only the differences since the previous publication, or what is set as default state using XCAP, is published following the extension for partial presence [76]. These bandwidth savings are important since a complete service presence information document for an environment with several services could become large.

4.5.2 Service presence information document

The service presence information document is a PIDF document (see 2.3.2 above [60] with extensions for service specific information. In these documents are services considered as presence information tuples. Extensions include supported actions, parameters, model number and type.

Moreover, because PIDF allows multiple extensions to be used simultaneously, service presence information can be extended to

include aditional presence information as well. For example GEOPRIV [62] could be used to include geographic information of where services are located and the rich presence extensions [61] could be used to add icons and describe the location where the service is located.

A binding from UPnP to service presence has been developed which can be found in 6.3Appendix C. Other bindings, such as for the Bluetooth service description format could be specified as well. Future studies could try to define a generic service presence description format for all service discovery protocols, or define functionality to allow different formats to interoperate.

Each service discovery gateway maintains one presence information document for all of their services[2]. This relationship can be seen in sequence 1 in Fig. IV-3, where the service discovery gateway's SIP Uniform Resource Identifier (URI) is included in the request-URI for the PUT request.

4.5.3 Subscribing to receive presence information

Service discovery gateways subscribe to the SIP URI of other service discovery gateways to discover their services. An example is given in sequence 2, "Subscribing to service presence", in Fig. IV-3. The current presence state is returned to the subscribing service discovery gateway from the presence server.

[2] Assuming that a service discovery gateway operates with only one SIP URI. For instance, in IMS a subscriber could be provisioned with several IMPUs that are SIP URIs.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<filter-set xmlns=”urn:ietf:params:xml:ns:simple-filter”>
 <ns-bindings>
  <ns-binding prefix=”pidf”
urn=”urn:ietf:params:xml:ns:pidf”/>
  <ns-binding prefix=”utl” urn=http://schemas.uia.no/UTL/”>
 </ns-binding>
 <filter id=”1234” uri=”sip:home.sdg@myoperator.com”>
  <what>
   <include type=”xpath”>/pidf:presence/pidf:tuple/upnp-root-
device/device/deviceType[urn:schemas-upnp-
org:deviceType:MediaServer:1]</include>
  </what>
  <trigger>
   <changed from=”CLOSED”
to=”OPEN”>/pidf:presence/pidf:tuple/pidf:status/pidf:basic</c
hanged>
  </trigger>
 </filter>
</filter-set>
```

Fig. IV-4 Subscription filter for being notified about media servers when they are switched on.

After the presence information has been updated, the subscribing service discovery gateway will be notified about the update by the presence server. An example is given in the last sequence in Fig. IV-3, “Service presence update”.

Subscribe filters, introduced in chapter 2.3.2, can be used by service discovery gateways to restrict notifications to particular service types and the frequency of notifications to be sent by the presence server. Fig. IV-4 describes an example filter that receives notifications only from services of type ‘media server’ and when their status changes to ‘online’.

Depending on the use case, service discovery gateways use the presence information differently. As an example, for remote service

control service discovery gateways could parse the service presence information and transform it into a native service description format. Then the native service description could be used together with an API to control the remote services. This technique is demonstrated in Chapter V.

4.6 Access control for service presence

Two methods are available to restrict access to service presence information. The simplest method is that a service discovery gateway only publishes presence information for services selected by an administrator. Thereby services that are not intended to be used remotely are excluded and the presence information document size is reduced. An implementation of this method can be found in the feasibility study described in chapter 5.3. For service presence information published, presence access policies [77] can be set by its service discovery gateways to grant and deny access to the presence information. The rest of this section focuses on this approach.

Service discovery gateways set their presence access policy using XCAP, as shown in the first sequence of Fig. IV-5, "Set presence access policy". An example of the initial state of such an access policy document for service presence is given in Fig. IV-6. This policy sets the subscription state for all new presence watchers to *pending*. Basically, this state means that the watcher won't receive any state information until the subscription state has been changed. For service discovery gateways to be able to change the policy from pending to *allow* or *block* they need to be aware of presence watchers. Therefore, before publishing presence information service discovery gateways subscribe to the watcher information [78, 79] for the presence event with the presence server (i.e., the event package 'presence.winfo'). This is shown in the second sequence of Fig. IV-5, "Subscribing to watcher information".

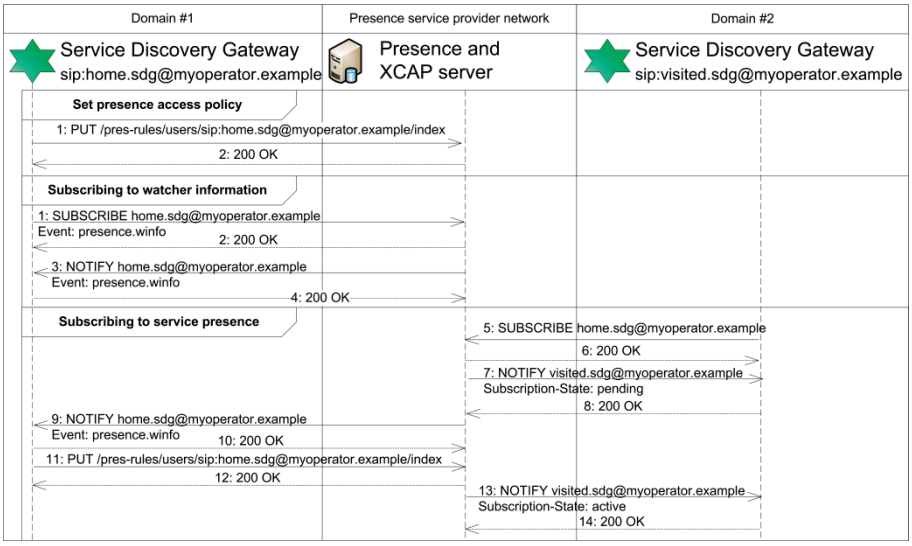


Fig. IV-5 Service presence access control sequence.

The presence server sends a NOTIFY request to the service discovery gateway when it receives subsription requests. This procedure is shown in the third sequence of Fig. IV-5, “Subscribing to service presence”. It is implementation specific how a service discovery gateway handles this notification. For instance, on a mobile phone a service discovery gateway could alert the user and ask whether the request should be allowed or blocked, similar to how incoming phone calls are handled.

After the user or the administrator has granted or denied permission to the request, the service discovery gateway must update the presence authorization rules. In particular, it should manipulate the policy for the sub-handling action for the requesting identity to *allow* or *block*, respectively. In addition to the normal block state a *polite-block* is also

```
<?xml version=”1.0” encoding=”UTF-8”?>
<cr:ruleset
 xmlns=”urn:ietf:params:xml:ns:pres-rules”
 xmlns:pr=”urn:ietf:params:xml:ns:pres-rules”
 xmlns:cr=”urn:ietf:params:xml:ns:common-policy”>
 <cr:rule id=”a”>
  <cr:conditions>
   <cr:many/>
  </cr:conditions>
  <cr:actions>
   <cr:sub-handling>confirm</cr:sub-handling>
  </cr:actions>
  <cr:transformations>
   <pr:provide-all/>
   <pr:provide-unknown-attribute ns=”urn:aml.uia.no:service-
presence” name=”upnp-root-device”>true</pr:provide-unknown-
attribute>
  </cr:transformations>
 </cr:rule>
</cr:ruleset>
```

Fig. IV-6. Example presence authorization ruleset for service presence. Sets the default policy to “confirm” for all watchers.

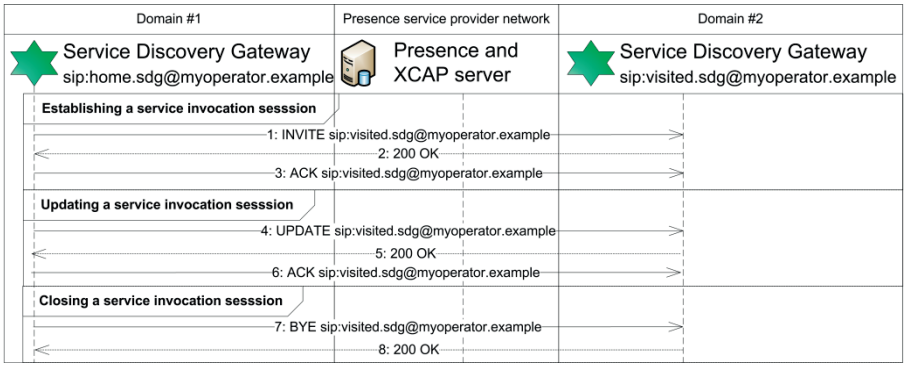


Fig. IV-7 Sequence diagram for the remote service control protocol.

specified. The difference between a block and a polite-block is that the former makes the subscriber explicitly aware of being blocked, and could reveal details of the policy being used. The latter option allows trying to trick desiring subscribers into believing they are subscribed to the service discovery gateway, but actually send them false presence information instead. For example, fake service information could be generated and included as presence information to it.

4.7 Remote service control

The remote service discovery protocol enables service discovery gateways to be aware of each other's services. In this section, a SIP-based protocol for controlling remote services is described that uses the service presence information. This protocol was originally proposed in paper [I]. Three stages make up this protocol: session establishment, session update, and ending a session. Sequences for these stages are shown in Fig. IV-7 and explained in the following sections.

Control requests can be sent from a service discovery gateway to remote services through the media plane after a session has been established.

4.7.1 Service control session establishment

To initiate control sessions, service discovery gateways send a SIP INVITE request to a peer, as shown in the sequence "Establishing a service invocation session" in Fig. IV-7 Following the offer/answer model [80] the body of the request should contain an offer in the Session Description Protocol (SDP) [81]. An example offer is given in Fig. IV-8.

Media-descriptors with the media type *"application/piranha"* in the offer specify services (e.g., UPnP device) that are offered[3] to be controlled. A new attribute, *udn*, is introduced for that. The value to be used here is obtained from the service presence information document. Moreover, the media descriptor specifies to use the Transmission Control Protocol (TCP) [36] as the transport protocol for control requests. Following [82], port 9, the discard port, is specified because the offerer does not want to receive incoming connections. It is also indicated that the offerer will create a new TCP connection in the *setup* and *connection* attributes.

[3] The term *offered* is a little odd here, because it is actually a request. Nonetheless, this term is used here to be consistent with [80].

```
1  v=0
2  o=visited.sdg 3380446179 3380446179 IN IP4 192.168.168.31
3  s=-
4  c=IN IP4 192.168.168.31
5  t=0 0
6  a=recvonly
7  m=application 9 TCP piranha
8  a=udn:uuid:9afb3231-345a-4cd1-b448-8866b79ff91b
10 a=setup:active
11 a=connection:new
```

Fig. IV-8 Offer for a remote service control session.

If the remote service discovery gateway accepts the offer, or parts of it, it adds port bindings for each accepted media descriptor at the local network gateway and sends a SIP response to the originating service discovery gateway (step 2 in Fig. IV-7). The SDP-answer included in the response is similar to the offer, but should specify the destination port for the TCP-connection to be used to control each service requested. Also like the offer, it should specify to use new connections (setting the connection-attribute to *new*). However, it should specify that it wait for incoming connections by setting the *setup* attribute to passive and use a new connection.

After receiving the answer in the response, the originating service discovery gateway then checks if it is acceptable. If acceptable, it sends an ACK-request to confirm it (step 3 in Fig. IV-7). Otherwise, it should either send a CANCEL-request to terminate the session or alter the session by sending an UPDATE-request with a different offer.

4.7.2 Updating service invocation sessions

During session lifetime, it could be necessary to alter the session description. For example, a service discovery gateway could desire to

control another remote service and therefore wish to add another media-descriptor to the session description. Another example could be to add another media-descriptor for receiving other media as part of the session.

Service discovery gateways update sessions by sending new INVITE-requests for the existing dialog, or if supported an UPDATE-request [83]. An example of the latter is shown in the second sequence of Fig. IV-7, "Updating a service invocation session". Otherwise, this procedure is the same as in initiating a new session.

4.7.3 Closing the service control session

Service discovery gateways close a session when it is no longer needed. For example, the requesting service discovery gateway could be finished controlling a remote service, or the service is no longer available and its service discovery gateway therefore decides to close the session.

SIP-sessions are closed with the BYE-request method, which is shown in the last sequence of Fig. IV-7, "Closing the service invocation session". All resources associated with the control session must be released when the session closes, including port mappings setup during the course of events of the session.

4.8 Virtualization of remote services

The service virtualizer function works together with service discovery gateways. Service discovery gateways inform service virtualizers with the presence of remote services that it receives from the presence

server. The service presence information is used by the service virtualizer to create local, virtual instances that are seen by legacy service controllers. Thereby this function enables legacy service controllers to control remote services as well.

Service virtualization offers optimizations when more than one service controller within a domain controls the same remote service. In for example UPnP control points have to subscribe with a service to be notified when events occur in the service, such as a state change. Because the local service controllers subscribe to the service virtualizer, and not directly to the remote service, can the service virtualizer function as an event aggregator. This means that the service virtualizer could subscribe once with a remote service, and when it receives a notification all locally subscribed service controllers are notified.

Another optimization that could be offered by a service virtualizer is to avoid control requests to retrieve state information. Often control requests are used to simply retrieve state information, instead of actually changing the state or execute other action. For example, media player controllers typically would like to receive state information such as the audio volume level. Since such a request will not change any state information, preferrably the service virtualizer could reply to the request instead of involving the remote service. However, the challenge to realize such a state cache is for the service virtualizer to know whether a request changes state information or not. Such semantics are not included in UPnP service description formats, which therefore would have to be altered for this to work.

In addition to providing remote service support for legacy applications, service virtualization allows decoupling of the remote service functionality in all service control applications. As a result, service control applications do not need to distinguish between local and remote services, except if it is required by the application to highlight this difference. This exception can be supported by adding meta-data to the local service description. For instance, in UPnP this could be achieved by adding an extra element to the device description, which should be ignored by legacy applications.

4.9 Service discovery gateway deployment targets

Based on the use cases given in Chapter III, this section describes several options for where the new service discovery gateway function could be deployed.

4.9.1 Residential gateways

As part of residential gateways, it could offer a whole LAN remote service discovery and control support. By virtualizing select remote services, any authorized client in the LAN could use them. Furthermore, when residents are away from their home their residential gateway's SDG allows them to access their services at home.

4.9.2 Mobile terminals

Deploying the service discovery gateway function at mobile terminals, such as mobile phones, enables the terminals to combine content and services from remote networks with locally available ones.

Remote services could be virtualized at the loopback adapter to make the remote services only available to the terminal itself. Thereby the service discovery gateway could be integrated in the platform and not with each application that executes on top of its platform.

Scenarios for mobile terminals carrying the service discovery gateway function include visiting infrastructure-based networks and creating ad-hoc networks on the go.

As car infotainment systems are evolving and becoming more powerful, they also will have the necessary resources to host service discovery gateway functionality themselves. For instance, with this new capability car infotainment systems could access remote media content from the driver's home and what offered by content providers.

4.10 Solution alternative

4.10.1 Utilizing the Atom publishing protocol

In [20, 84] P. Belimpasakis et al. proposes a solution that utilizes the Atom publishing protocol [85] and the Atom syndication format [86] for remote service discovery, primarily targeting UPnP.

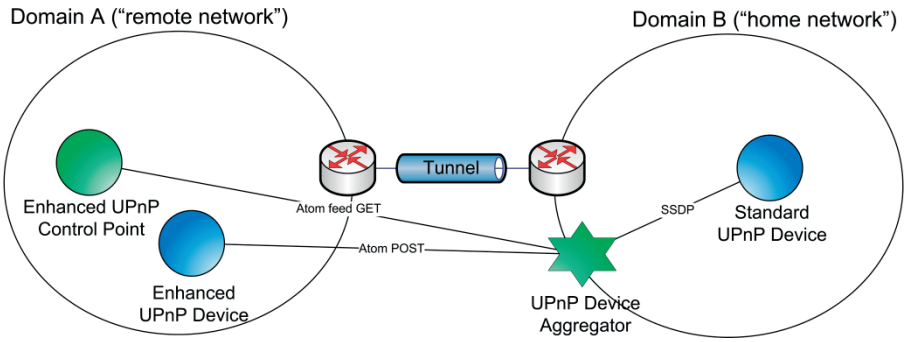


Fig. IV-9 Architecture of the Atom-based solution.

The architecture of this solution is shown in Fig. IV-9 (based on Figure 2 in [84]). This solution assumes that a secure Virtual Private Network (VPN) tunnel has been established between two domains. SSDP information is syndicated to a new function named UPnP Device Aggregator in the remote domain. This function is responsible for multicasting the received SSDP information in the local domain to the UPnP multicast group. Local control points handle these SSDP multicasts as any other SSDP multicast.

4.11 Solution evaluation and comparison

In this section, first the service presence solution and the reference Atom-based solution are evaluated based on the solution requirements given in Chapter III. The last section of this section compares these solutions against each other based on the evaluation results.

Rating score values used are plus ('+'), null ('o'), and minus ('-'). Here plus means that the requirement is satisfied, and minus the opposite. Null means that the requirement is not directly satisfied, but not unsatisfied either.

4.11.1 Atom-based solution

Table IV-1 Evaluation of functional requirements for the Atom-based solution.

Requirement	Evaluation	Score
Remote service discovery for ubicomp environments	Supported.	+
NGN services aware of services in ubicomp environments	Not explicity mentioned in [84]. But no restrictions to deploy the UPnP Device Aggregator function in a NGN service provider network.	+
Remote service control	Supported.	+

Table IV-2 Evaluation of quality requirements for the Atom-based solution.

Requirement	Evaluation	Score
Compatibility	Requires enhanced control points and devices that operate in remote domains.	-
Security	Depends on secure VPN tunnels for security.	o
Integrity	- Supports roaming users. - No mechanism provided to handle that the VPN tunnel terminates, which would cause integrity issues.	o
Scalability	VPN does not scale well because it requires careful administration of IP addresses and subnetworks. Especially considering NGN service providers that should handle thousands, or even million, concurrent sessions is this difficult to achieve.	-

4.11.2 Service presence based solution

Table IV-3 Evaluation of functional requirements for the service presence based solution.

Requirement	Evaluation	Score
Remote service discovery for ubicomp environments	Supported.	+
NGN services aware of services in ubicomp environments	Supported.	+
Remote service control	Supported, although the solution itself does not handle private IP addresses in messages sent across domains very well. Utilizing "Home DNS" [87] should improve this.	o

Table IV-4 Evaluation of quality requirements for the service presence solution.

Requirement	Evaluation	Score
Compatibility	Requires a service discovery gateway in each domain. Supports all service types. With service virtualization the solution is also compatible with legacy service controllers.	+
Security	Supports access control for service presence.	+
Integrity	- If service discovery gateways crash, the integrity will be affected until the presence information lifetime expires. - Roaming services and service controllers are supported.	o
Scalability	- Presence server offloads service discovery gateways from notifying all services. - No design limitations to the number of concurrent sessions.	+

4.12 Comparison and analysis

Table IV-5 Comparison of remote service discovery and control solutions.

	Requirement	Atom-based	Service presence
Functional requirements	**Remote service discovery for ubiquitous computing environments**	+	+
	NGN services aware of services in ubiquitous computing environments	+	+
	Remote service control	+	o
Quality requirements	**Compatibility**	-	+
	Security	o	+
	Integrity	0	o
	Scalability	-	+

The comparison shows that both solutions fulfill the functional requirements, with the Atom-based solution better in remote service control. However, the Atom-based solution scores low on the quality requirements. Therefore, the Atom-based solution seems most appropriate to be used for within a small group, like a family or a few friends, and not together with NGN service providers. On the other

hand, the service presence solution high scores on the quality requirements means that it is capable of fulfilling also the high demands of service providers.

It should be taken into consideration here that the Atom-based solution was not designed to fulfill all requirements that are used here. Being independent of NGN makes that solution easier to be deployed. Combining the features and advantages of both solutions would create a more powerful and flexible solution, such as the Atom-based solution's better remote service control support together the scalability of the service presence-based solution.

Chapter V

Feasibility studies

This chapter describes feasibility studies accomplished and findings from these. Studies covered are how UPnP works on mobile phones, remote service discovery and control, virtualization of remote resources and services, media delivery from service providers to ubiquitous environments and portable gateways in ad-hoc environments.

5.1 ONE Portable Player: UPnP on Mobile Phones

5.1.1 Objective

The main objective of this study of UPnP on mobile phones was to get an initial experience with UPnP as a wide spread example service discovery protocol and to find out if and how mobile phones can use it. Of particular concern was whether mobile phones (in the year 2005) can handle the large, complex XML documents that are used in UPnP.

Moreover, we investigated whether commercial off-the-shelf UPnP appliances actually support external control points. Such support is a core assumption in the remote service discovery and control protocols. This study was based on the vision of a future media portal given in [63]. It was conducted in two stages. In the first stage, a mobile control

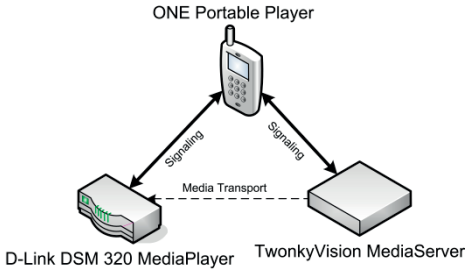


Fig. V-1 Logical view of the prototype along with test devices, a media player and a media server.

point was created and in the last stage hosting of UPnP devices were added to the prototype. Further details of stage 1 can be found in [88].

5.1.2 Prototype stage 1: mobile multimedia control point

This prototype consists of an application that controls UPnP AV devices (see 6.3Appendix A for further details) for mobile phones. Because UPnP requires IP-network support, two mobile phones equipped with WLAN were selected as hardware for the prototype. In particular, a Qtek 9100 (inbuilt WLAN) and a HP 6515 with a Socket SDIO WLAN card were used. Fig. V-1 depicts the prototype, running on a mobile phone, together with test devices.

To allow the application to be used on different mobile phone systems the platform defined by the Java™ Technology for the Wireless Industry (JTWI) [89] specification was selected. This platform is based on the Java Platform, Micro Edition (Java ME)[4], consisting of the Connected Limited Device Configuration (CLDC) 1.1 [90] and the Mobile Information Device Profile (MIDP) 2.0 [91].

[4] Java ME was formerly known as Java 2 Platform, Micro Edition (J2ME).

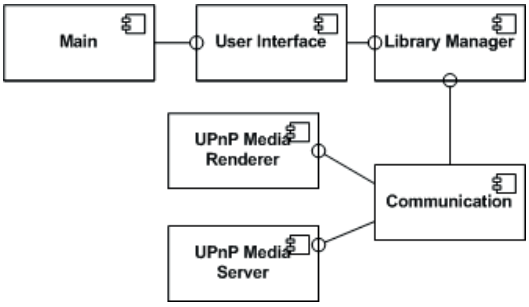


Fig. V-2 Component view of the ONE Portable Player.

The application consists of six components, as shown in Fig. V-2. The Main component is the main entry point for the application, and is responsible for starting the other components. Moreover, it keeps track of configuration settings for the application.

The User Interface component handles all interaction with the user. It provides a menu-driven user interface that allows users to select media from any UPnP Media Server available and play selected content at any available UPnP Media Renderer. Screenshots are given in Fig. V-3. For example, photos taken by a camera can be shown on a TV. Moreover, it

(a) (b) (c) (d)

Fig. V-3 Screenshots of the user interface. From upper left to lower right: (a) main menu of the application, (b) browsing of a media server, (c) play screen and (d) media server discovery.

includes a ”Play screen” to control media rendering as shown in Fig. V-3(c).

Furthermore, the Library Manager component gives an abstraction to content, and is used by the User Interface component to handle local and remote media in the same manner.

The Communication component takes care of all low level communication tasks. For example, it connects to access networks and handles service discovery and invocation of SOAP actions. Other components such as the User Interface can register to receive service presence updates. Moreover, it gives base classes for UPnP device and service control. Two sub-components, UPnP Media Server and UPnP Media Renderer, extend the UPnP base classes to control their respective device type.

5.1.3 Prototype stage 2: phone based media server and renderer

In this stage main focus was on adding support for hosting of UPnP devices. A UPnP Media Server and a UPnP Media Renderer, named Local Media Server and Local Media Rendererm respectively, were

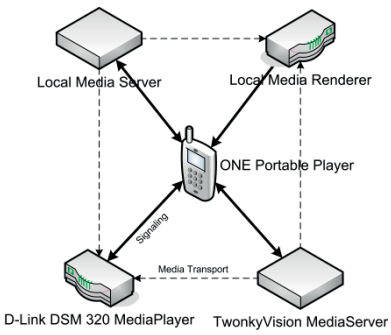


Fig. V-4 Logical view.

created using this support. The former makes content at the mobile phone available to be used with media renderer devices, and the latter uses the phone as a media renderer. As shown in Fig. V-4, these new features allow users to combine resources on the phone with external UPnP resources.

The UPnP device hosting support component was designed to completely take care of UPnP phases 0, 1 and 2 for all devices and services. In addition, for the control phase it will take care of handling SOAP for the services. Hence, device and service implementations can focus on the application specific parts instead of UPnP details. A structural view of this component is depicted in Fig. V-5. Devices, like the Local Media Server and the Local Media Renderer, implement the UPnPHostingDevice interface or extend from the UPnPHostingDeviceBase class that gives default handling. Classes that implement the UPnPHostingDevice interface can be registered with the UPnPHosting class.

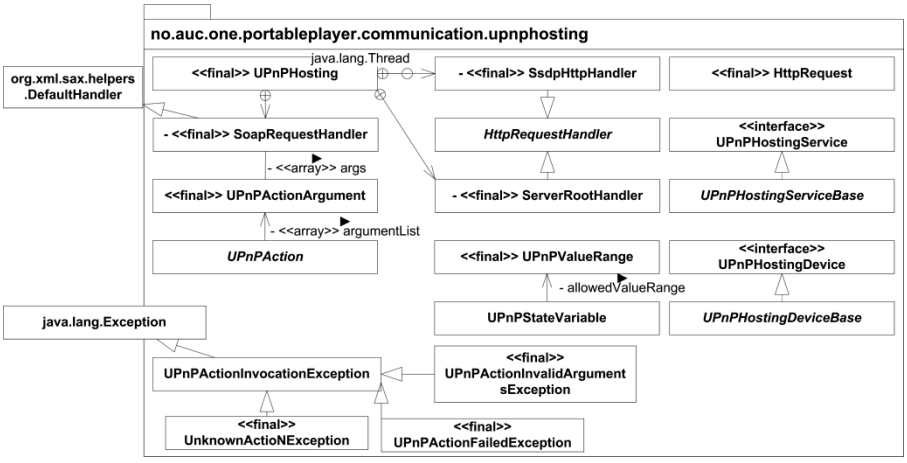


Fig. V-5 Structure view of the UPnP Hosting Support component.

When a device is registered for hosting, the UPnPHosting class generates XML description documents by retrieving the information needed using the UPnPHostingDevice interface. Services follow a similar pattern, but are added to a hashtable in the device class that can be retrieved by UPnPHosting. Furthermore, the hosting class will setup a timer to repeatedly send out SSDP NOTIFY requests and reply to SSDP M-SEARCH requests for the device. Incoming SOAP requests are invoked with an associated class of a service that extends the UPnPAction class. Faults that occur when handling such SOAP requests are signaled by throwing the UPnPActionInvocationException class, or one of its descendants.

A lightweight HTTP server implementation is included in the UPnP hosting support. This server is intended to host the description documents, receive SOAP action requests for the hosted services and to support receiving GENA requests. To handle these different web applications the server has introduced a concept called namespaces that allows different Java classes to handle different requests. A namespace is defined as the first segment of the Request-URI path-component in a HTTP request. For example, the namespace of `http://onepp/control/mediaserver` is 'control'. It is up to the namespace handler how to interpret the rest of the Request-URI. A namespace handler is registered with the server to handle a specific namespace. When the server receives a request it invokes the associated namespace handler.

A namespace handler called "Web Control Point" was implemented to provide non-UPnP capable clients to control UPnP AV devices. Control

is achieved through a set of simple web pages that are generated based on currently available UPnP devices.

5.1.4 Findings

Several issues were encountered when developing the prototypes, although in the end it turned out to be feasible on the target platform.

CyberLink for Java [92] is a free, open-source UPnP stack for Java. However, it does not support J2ME CLDC that was selected as the target platform. Because it turned out to have too many incompatibilities to port CyberLink for Java to J2ME CLDC, it was decided to implement a new UPnP stack instead. A major problem faced with UPnP on J2ME CLDC is that the platform does not support the Internet Group Management Protocol (IGMP) [93]. Therefore, multicast groups cannot be joined, which SSDP depends on. However, because sending multicast messages works this issue can be worked around to some extent. For example, can SSDP M-SEARCH requests to query for available devices or services. However, applications will not be able to receive SSDP NOTIFY requests to keep track of services. Therefore, device state inconsistencies could easily occur, such as not being aware of devices that become unavilable[5]. This limitation degrades the user experience because users may try to control a device that is no longer available.

Hosting devices without IGMP support is more restricted since it will not receive SSDP M-SEARCH requests from control points. It can only

[5] Informed with a SSDP NOTIFY request with the notification sub type (NTS) set to “ssdp:byebye”.

be discovered by control points receiving SSDP NOTIFY requests that are sent. Since UPnP specifies that 30 minutes is the minimum time between sending such requests it could take some time to discover the hosted devices. The UPnP Low Power Proxy [94, 95] offers a workaround for this problem, however. A Low Power Proxy can take responsibility for handling service discovery and description for other devices, so that these devices can sleep and conserve energy resources. Residential gateways could provide a good platform to host such a proxy. To verify that this works as A subset of [95] was implemented to verify that this works.

A related issue to multicasting is that UDP support is optional in MIDP 2.0 [96] (see Table 1 and Table 2). Qtek 9100 ships with the TAO intent JVM[6] and does not include UDP support. Therefore, the IBM Web Everyplace Micro Environment (WEME) with its J9 Virtual Machine (VM) was used instead.

However, several interoperability issues with the HTTP implementation in J9 VM were found when sending and receiving messages with the UPnP test devices. Because it was not possible to modify the HTTP stack, the workaround was to develop a new HTTP stack for J2ME CLDC. This stack was also necessary to implement the HTTP server since JTWI does not include any web server support.

Furthermore, it was discovered that the SOAP support for J2ME, provided by the optional package "J2ME™ Web Services" [98] could not be used. This package specifies a static remote procedure call API for

[6] Tao Group that produced Tao intent does no longer exists and their portfolio was apparently sold to a venture capitalist company [97].

invoking web services. Two reasons were identified for why it cannot be used. First, it depends on using Web Service Description Language (WSDL) [99] documents for services it shall be used with to generate static stubs at compile time. Perhaps this could be workaround by transforming UPnP description documents into WSDL documents. The second reason is that the reference implementation used generated SOAP messages that were not accepted by some of the test devices, most probably because different XML namespace [100] names are used then in the UPnP Device Architecture are used. In addition, the SOAP stack includes some XML attributes that are not used in UPnP. Supposedly, these test devices have been designed to strictly follow the UPnP Device Architecture, including examples given within, and therefore do not interoperate with these SOAP messages. A custom SOAP library was implemented to work around this issue as well.

There are many incompatibilities on mobile phones, as they do not meet basic Internet functionality. Except all these interoperability problems, the application could discover and control the test devices.

It was found that processing large XML documents would take a considerable amount of time, from when it was sent from the test device and to it was parsed. However, it was found that the parsing itself did not need much time. Closer investigation indicated that the bottleneck is actually the memory bus of mobile phones, which typically is just 100-200 MHz. Therefore, it takes some time to transfer the data received to memory before parsing can start.

The action invocation design in UPnP Hosting Support leads services to include several small private classes, one for each *action* supported. These classes are necessary to be able to both get information about actions to generate service description documents and to invoke methods through reflection. A large amount of small classes is not optimal for two reasons. First, it gives a bad development experience, since it requires a lot of extra. The second reason is that it becomes quite a lot of extra classes that the class loader will have to process and load into memory. For example, the AVTransportService of the Local Media Renderer device has 13 actions and therefore 13 private classes that descend from the UpnPAction class. With better reflection support it could have been redesigned so that each action becomes a simple method of a service instead.

The web control point feature seeded a master thesis [101] that investigated how to allow thin clients to use IMS services, such as presence and messaging, through web applications.

5.2 Remote service control

5.2.1 Objective

This study investigated whether the remote service control protocol works as expected. One aspect was whether local and remote services could cooperate, for example a local media player and a remote media server. Another aspect was to see whether media streaming from a private network to another private network works, and the perceived quality of experience of remote media access.

Furthermore, it was desirable to see if a package could be developed that could be used at both residential gateways and mobile phones.

5.2.2 Prototype description

This prototype extends the ONE Portable Player with remote service control. It focuses on the remote multimedia access use case described in chapter 3.1, and allows controlling and using content from a remote media server with local media renderers.

The earlier developed prototype was refactored into a library, named xIGA, and an application for mobile phones, named Portable IMS Gateway (PIGA). The Main and User Interface components were moved to PIGA, which then depends on xIGA for all other functionality. A new application, named Home IMS Gateway (HIGA) for Remote Access (HIGA-RA), was created that also depends on the xIGA library. HIGA-RA was developed to be deployed at residential gateways and similar embedded systems. Therefore it does not include any graphical user interface but rather a tiny console interface.

A limitation with the prototype is that HIGA-RA can only terminate remote service control requests, and PIGA can only originate such requests.

Due to the lack of proper multicasting support in J2ME CLDC it was decided to port the PIGA and HIGA-RA applications along with the xIGA library to a more feature rich J2ME platform. The new platform is made up by the Connected Device Configuration (CDC) [102] with the Personal Profile (PP) [103]. Besides multicasting, this platform offers richer reflection support that could be used to redesign the UPnP control support of the UPnP Hosting Support Framework.

In addition to refactoring the previous prototype, three new components were added to xIGA. These components are named Internet Gateway Device, PIRANHA, and Remote Media Server and are indicated with a star (*) in Fig. V-6. The Internet Gateway Device component provides control of the Internet Gateway Device DCP.

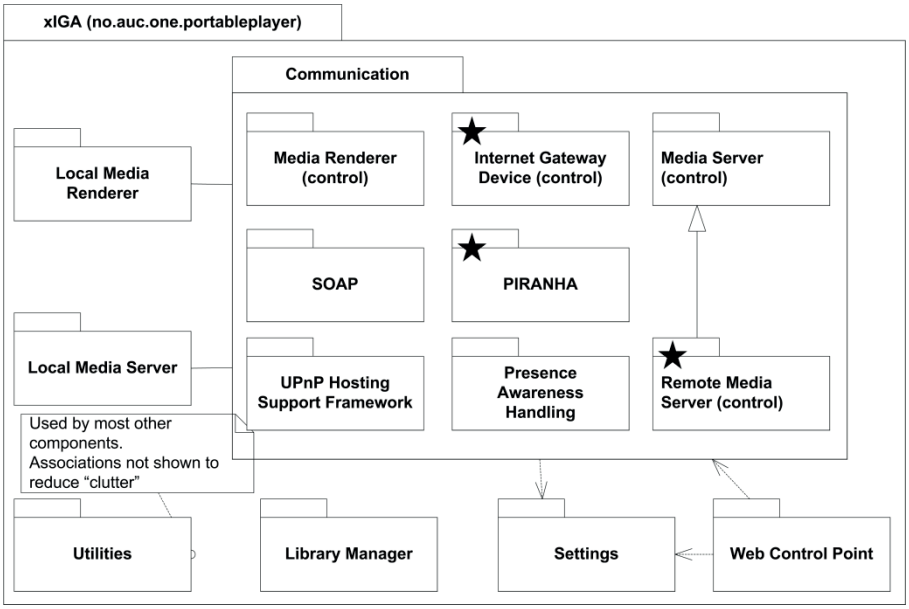


Fig. V-6 Component view of the xIGA library in the remote service control prototype. A star (*) marks new components in this prototype.

PIRANHA implements the remote service control protcol, which is described in chapter 4.6. SIP support is provided by the NIST JAIN SIP [104-106] stack. Two classes, PiranhaSessionManager and PiranhaSessionHandler manage originating and terminating requests for remote service control, respectively. PiranhaSessionHandler uses the new Internet Gateway Device control component to create Network Address Translation (NAT) [107]-bindings with the local gateway.

Finally, the Remote Media Server component extends the Media Server control component to handle private IP addresses in the received content directory information. Thus, when receiving Browse action responses it changes the host and port part of content URIs to the remote address and port, which is known from the SDP-answer in the session initiation response.

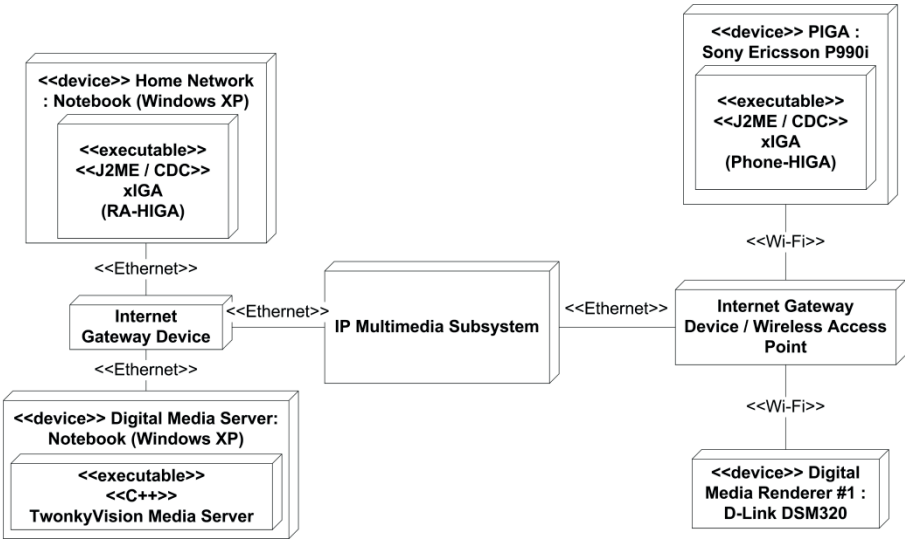


Fig. V-7 Deployment view of the remote service control prototype.

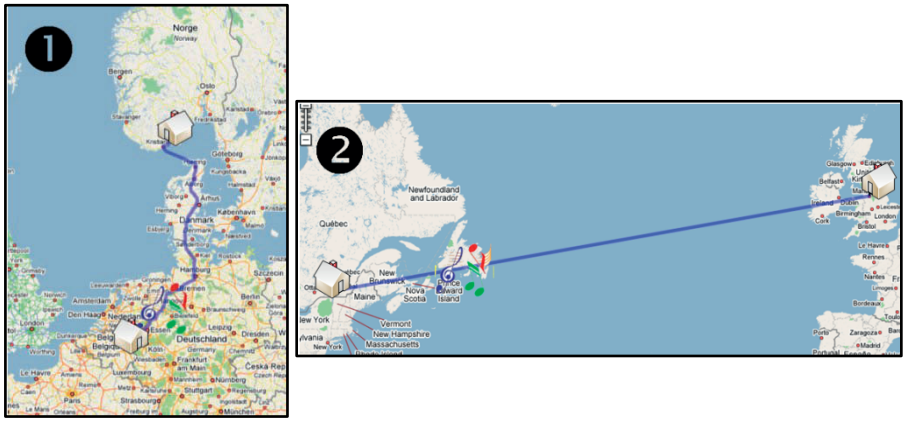


Fig. V-8 Long-distance testing of remote service control and media delivery. In (1) music was streamed from Ericsson Eurolabs Deutschland in Aachen, Germany to the University of Agder in Grimstad, Norway. And (2) shows a test case were pictures were downloaded from Ericsson Canada Inc in Montreal, Canada to EMCC Software Ltd, Manchester, United Kingdom.

Three different deployment setups were used to test the prototype for the set objectives. Fig. V-7 depicts a reference visualization of these setups. In the first, and simplest, configuration all nodes were deployed in the same lab, and therefore unable to indicate any delays introduced from the remote media delivery. The other two configurations were deployed in different geographical locations, as shown in Fig. V-7. Both of the tests were accomplished without significant delays, even transatlantic. Although music streaming and still pictures both worked fine, video streaming does was found to not work well. This is due to the TCP flow control mechanism because HTTP is used as transport protocol.

5.2.3 Findings

This study shows that the remote service control protocol works well without significant delays, even during transatlantic testing. The exception is that video streaming was perceived with unacceptable

quality of experience due to problems with the TCP flow control mechanism. Probably, because HTTP is the only mandatory transfer protocol in DLNA, most of the current media servers only support that transfer protocol.

The JAIN SIP stack was found to be difficult to use for this purpose because it gives a very low-level perspective to SIP. A more high-level API, such as the IMS Services API [108], could therefore be more suitable for this application.

Handling private IP addresses by changing the message body in all messages exchanged between the service discovery gateways works, although it is a brute force approach to the problem.

5.3 Remote service discovery

5.3.1 Objective

In this study, the main objective is to study whether the remote service discovery protocol works as designed. Both PIGA and HIGA-RA should support service presence publishing and subscribing to service discovery gateways.

5.3.2 Prototype description

Remote service discovery required several additions to xIGA, PIGA and HIGA-RA. In addition, a lightweight SIP Presence server was developed that is named Serivce Presence Server (SPS) to add support for SIP Presence in the core network. In the following SPS is covered first, thereafter changes to xIGA, PIGA and HIGA-RA.

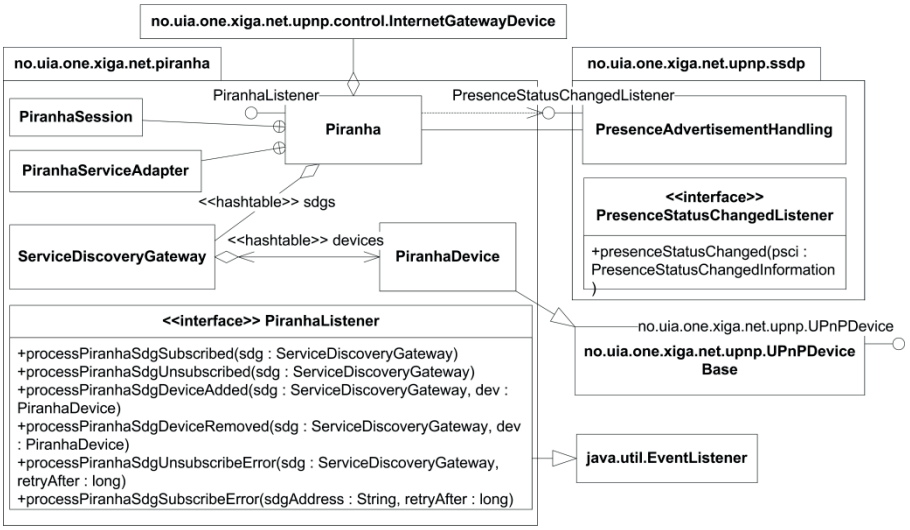


Fig. V-9 Structural view of the PIRANHA component in xIGA in the remote service discovery feasibility study.

Service presence server

SPS was created because there currently was no stable open-source presence server alternative available that would support IMS. Due to time constraints a lightweight presence server was developed as a SIP servlet [109], instead of trying to adapt existing open source alternatives. It handles PUBLISH and SUBSCRIBE requests and sends NOTIFY requests to presence watchers. This SIP servlet was deployed to a Sailfin application server [110] in the core network.

xIGA PIRANHA component

In xIGA the SIP stack was changed from JAIN NIST to the Ericsson IMS Client Platform (ICP) [111], which at the time this feasibility study was conducted was an early implementation of JSR281 (also known as pre-JSR281). Changing the SIP stack and introducing remote service

discovery required substantial changes to the PIRANHA component, and the final design of the component is shown in Fig. V-9.

At the heart of the PIRANHA component lays the Piranha class. It allows other components to subscribe and unsubscribe to service discovery gateways, retrieve a table of currently subscribed service discovery gateways, and make local devices available and unavailable for remote usage. To be aware of available local devices it implements the PresenceStatusChangedListener exposed by the Presence Advertisement Handling component. Furthermore, the Piranha class exposes the PiranhaListener interface to allow other components to receive events related to this functionality, such as when it has subscribed to another service discovery gateway or a new remote device has been discovered. For instance, this interface PIGA updates its user interface based on events received through the PiranhaListener interface.

To process IMS signaling, via ICP, Piranha uses the PiranhaServiceAdapter. For example, it processes presence update notifications of subscribed service discovery gateways. Moreover, remote service usage requests are also processed by this class. The PiranhaSession is used to take care of such requests, and inherits functionality from the former PiranhaSessionHandler class.

The ServiceDiscoveryGateway class represents a remote service discovery gateway and includes a table of available devices. Remote devices are instances of the PiranhaDevice class that extends the UpnPDeviceBase class to give a reference to its associated ServiceDiscoveryGateway, establish a remote service usage session

(similar to the old PiranhaSessionManager class), and close the remote service usage session.

Management user interface for HIGA-RA

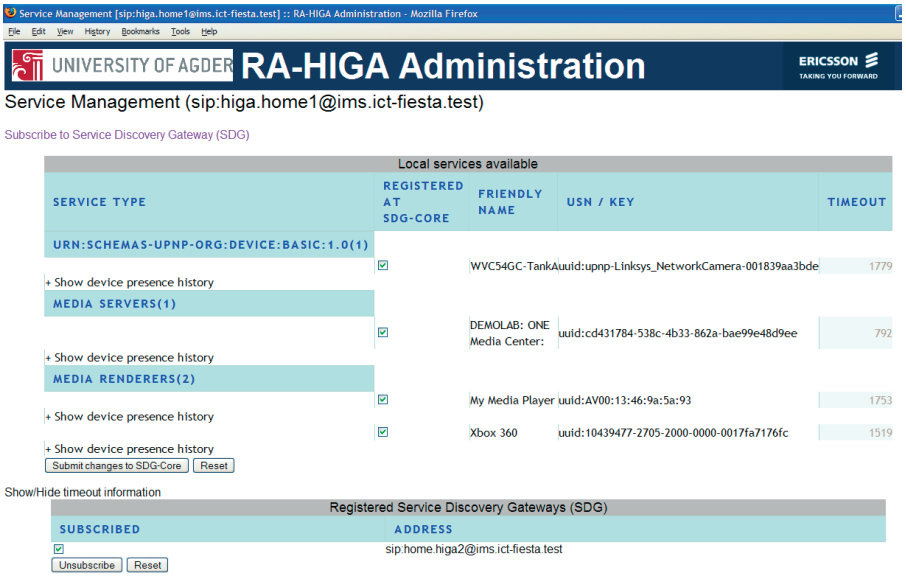


Fig. V-10 Screenshot of the web-based management user interface for HIGA-RA.

A web-base management user interface was created for HIGA-RA that is hosted by the embedded HTTP server. It is divided into two parts, where the top part shows local device state grouped by device type and let users select which devices to register (i.e., publish) with the presence server. The bottom part shows service discovery gateways and allows unsubscribing from these. A link at the top of the page allows users to subscribe to service discovery gateways.

Service discovery support in PIGA

Additional entries were added to the menu-based user interface to allow users to control remote service discovery. Similar to the web-

based management user interface, users can here also subscribe and unsubscribe to service discovery gateways and select which devices to make available for remote usage. Remote AV devices that are discovered can be used together with the existing AV control functionality, such as browsing a remote media server.

5.3.3 Findings

The study shows that remote service discovery can be enabled by the presence service. However, applications such as PIGA become more complex since they need to juggle between local and remote devices because it was infeasible to abstract away all the differences between a local and a remote device. In particular, a remote service usage session must be established before a remote device can be used.

Using a high-level IMS API eased implementation work, but naturally, it also introduced limitations that are not found with the more low-level NIST JAIN SIP stack used before. One gap found was that a body could not be added to SUBSCRIBE requests and therefore subscribe filteres could not be studied here. In addition, ICP was found to be difficult to be deployed to test devices because it is comprised of two parts, namely a library that exposes the API and a service that handles communication with the core network. This service is implemented in native code for a given platform, probably for performance reasons, and therefore it restricts which platforms that are supported.

5.4 Virtualization of remote devices

This feasibility study was based on work undertaken to provide socalled place-shifting for a joint-demonstration by Sony, Ericsson and

Sony Ericsson at the Broadband World Forum Europe 2007. UiA contributed to this demonstration as Ericsson's partner in the "ONE Project". Unfortunately, an external issue with the residential gateway used forced this work to be stopped.

Parts of this feasibility study was performed by Jesús Ruiz de Mier Gómez in his master thesis [112].

5.4.1 Objective

This study was conducted to see if the service virtualization concept, described in chapter 4.8, works and let an off-the shelf media renderer with integrated control point functionality control a remote media server.

What information is necessary to be obtained to create a virtual UPnP device, and whether this information could be made available to the service virtualizer was one aspect investigated as part of this study. Related to this aspect was whether a device's unique device name should be reused or a new unique device name should be generated.

The last aspect covered was how to handle private IP addresses as part of service virtualization. With service virtualization part of the communication is handled by external control points and therefore more challenging to handle than previously when all details were handled by xIGA.

Due to time constraints, only service discovery and control could be covered in this study. Therefore, event handling was left out for future study.

5.4.2 Prototype implementation

To demonstrate service virtualization the service virtualizer function described in chapter 4.8 was added to the xIGA library and used with HIGA-RA deployed at two residential networks. The new service virtualizer functionality adds two new classes, RemoteServiceVirtualizer and ControlProxyInvocationHandler, to xIGA, that are shown in Fig. V-11.

RemoteServiceVirtualizer implements the PiranhaListener interface and is registered with Piranha to receive events. When it is notified about new remote devices it will virtualize them. Ideally it should only virtualize selected devices, but that feature was left out in this study. The notification from Piranha includes a reference to a PiranhaDevice

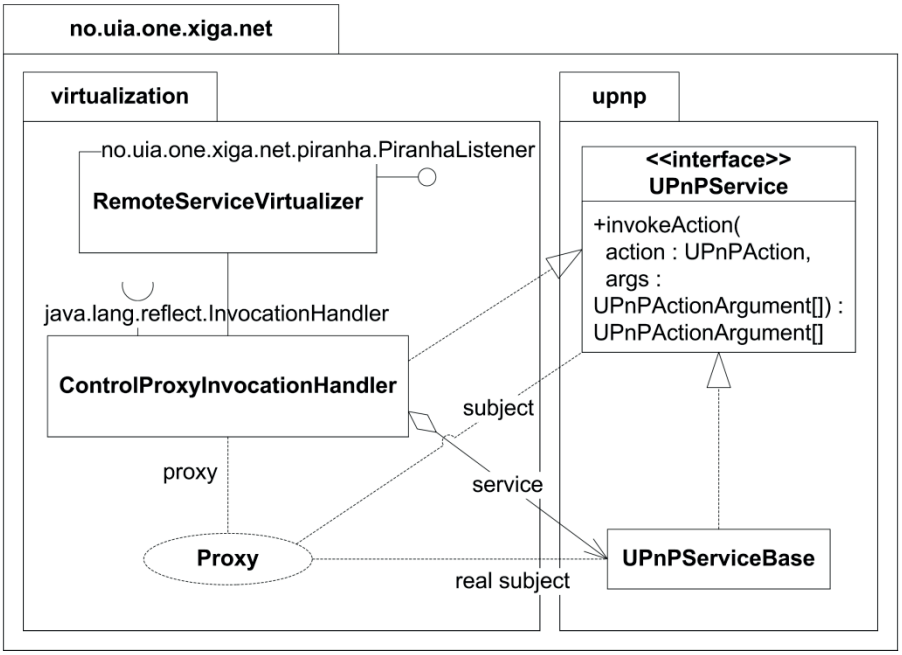


Fig. V-11 Structural view of the service virtualizer.

class instance that represents the remote device based on the service presence information. Remote devices are virtualized simply by hosting this PiranhaDevice instance by the UPnP Hosting Support Framework. Similar to other hosted devices, the UPnP Hosting Support Framework will announce them to let other control points discover them and include them in replies to service discovery queries. Also, description documents are generated and hosted at the embedded HTTP server. Similarly, SOAP control requests and event subscription requests will be received by the UPnP Hosting Support framework.

Because PIRANHA does not have any special control classes for handling remote requests, the same control classes that are used with local devices will be used with remote devices. However, if control requests include IP addresses, these may be private IP addresses and should be transformed to a public IP address instead. Previously in the Remote service control study (see chapter 5.2) a special component, Remote Media Server, was introduced to handle this problem. However, this approach was found to be difficult since the system must always differentiate between local and remote devices. Besides, usually the control logic is the same. Instead, before the RemoteServiceVirtualizer hosts the device it will inject the ControlProxyInvocationHandler for the service control classes. The ControlProxyInvocationHandler uses the dynamic proxy feature of Java's reflection support to pre- and post-process control requests for the remote device. As shown in Fig. V-11, this feature is based on the Proxy design pattern [113] where the ControlProxyInvocationHandler is

the proxy. Before responses are returned to the control point it will replace private IP-addresses with the public IP address of the device, which it learns from the remote service usage session information.

The solution was tested using the D-Link DSM 320 Wireless Media Player, an off-the-shelf UPnP Media Player, which controlled a virtualized remote media server located in a different network. In this test another off-the-self device, the Synology DiskStation 109j, was used as a UPnP Media Server.

5.4.3 Findings

To successfully virtualize devices a complete set of information about them and their associated services and embedded devices must be obtained. No issues were found by reusing the unique device ID of the device. Mobile control points can establish a relationship with a device in one network and maintain it when roaming since it can always be identified using the unique device ID. Also the unique device ID can be used to avoid virtualizing a device in the device's local network.

Using the control proxy avoids introducing special classes for handling remote devices. However, transforming each message is rather slow and the problem is solved more elegantly using P. Belimpasakis' "Home DNS" [87] solution instead that does not need to alter the messages.

In addition to enabling standard UPnP control points to use remote devices, it was also found to simplify remote control capable applications. Instead of being part of an application the remote service

discovery and control logic can be deployed as part of control points' underlying platform.

5.5 Media delivery to remote renderers controlled by the mobile phone

5.5.1 Objective

There were two main objectives in this feasibility study. The first objective was to understand how service providers could provide services to devices inside local networks like residential networks. Secondly, it investigated how feature phones, which do not have WLAN support, can establish relationships to their local environment.

5.5.2 Prototype description

Scenario

This study focuses on a scenario where users visit a hotel where the hotel rooms include a media player with broadband connection, such as in the use case described in chapter 3.1.1. An illustration of this scenario is given in Fig. V-12.

Users with WLAN-capable smart phones can discover and control the media player over WLAN. For users that carry feature phones that are not WLAN-capable it is not that simple since their mobile phones cannot directly control the media player over WLAN. Therefore, a new function, named residential control device, is introduced that uses proximity technology, such as barcodes and Near Field Communication (NFC), to allow such feature phones to indirectly connect to the environment.

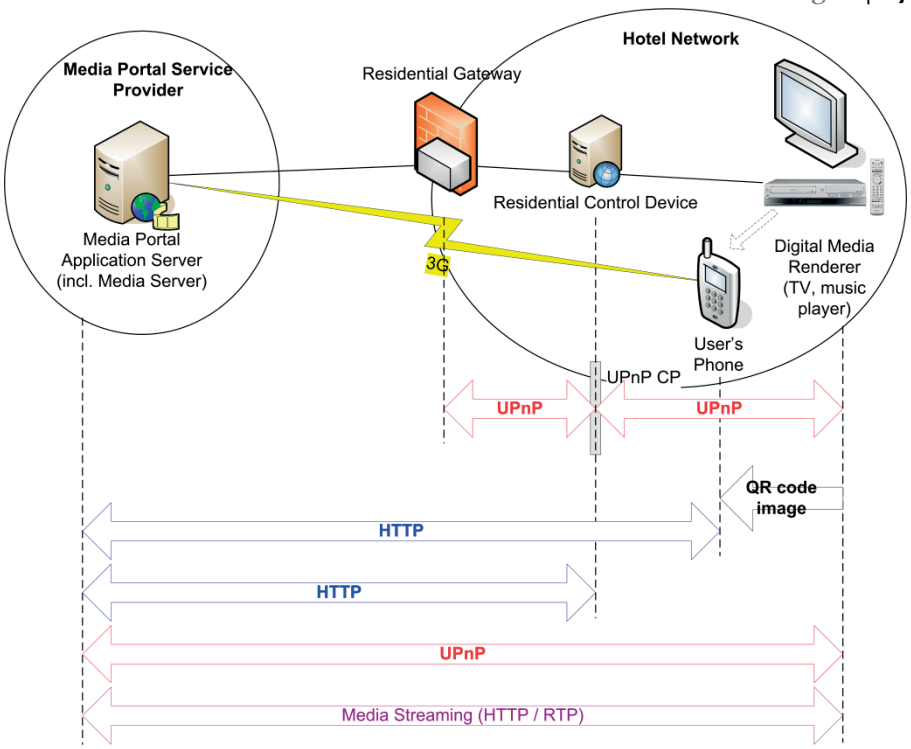


Fig. V-12 High-level system architecture and signalling flows of the media portal prototype, from paper [V].

Through the residential control device a reference to the environment can be obtained that can be transmitted to a third party. This reference let the third party communicate with the residential control device to discover available devices in the users' environment. Via the third party service and the residential control devivce users are then indirectly connected to their environment and can control it.

Implementation and deployment

In the following, a brief description of the five entities is shown in Fig. V-12 is given.

(a) (b) (c)

Fig. V-13 Media player prototype. (a) "Welcome screen" displayed at the hotel room TV. (b) Photo capture screen of the media player client application (white because the screenshot cannot include the video stream from the camera). (c) 2DQR decoding progress of the media player client application.

A residential gateway connects the hotel network to the Internet. Local entities can configure NAT-bindings by using the UPnP Internet Gateway Device (IGD) DCP (see chapter 6.3A.2.2).

The residential control device was developed as a Java Platform, Standard Edition (Java SE) application. Using the aforementioned xIGA library it discovers local UPnP devices and hosts a web application that is used to administer which devices are available in a room. Moreover, it generates an image that welcomes the user, so called "Welcome image" like Fig. V-13(a), is displayed at a selected media renderer. The "Welcome image" contains a two-dimensional bar code (QR code) [114] that encodes the environment reference. The reference is an http-URL that refers to the residential control device. Using a camera the QR code can be obtained and then decoded.

For this study, a Sony Ericsson W915i was used for mobile phone. It is a typical feature phone with a 2 megapixel camera. A simple Java Platform, Micro Edition (Java ME) application, named Media Portal

Client, was developed to take a snapshot (see Fig. V-13(b)) of the QR code and decode it (see Fig. V-13(c)). After decoding the URL, the Media Portal Client navigates to the Media Portal home page using the mobile phone's web browser and passes it the decoded URL.

The media portal, a Java Platform, Enterprise Edition (Java EE) web application, offers personalized content and media renderer selection. When clients connect to the web application the environment reference URL should be passed to the media portal. Next, the media portal connects to the residential control device and retrieves a list of available devices. After users have selected content this device list is presented to them to select the media renderer they would like to use. Then the media portal controls the media renderer to render the selected content from its streaming media server.

5.5.3 Findings

Services can be delivered to off-the-shelf consumer devices in users' environment, with users' terminal staying in control of service access and delivery.

For phones that cannot directly connect to the environment proximity technologies together with a third party service enables these phones to be indirectly connected to their environment.

One area for improvement is system responsiveness. The current Media Portal Client was found to be too slowand unreliable in decoding QR-codes. Future mobile phones will likely include native, or

even hardware, support for decoding QR-codes, which will improve these issues.

Moreover, latencies between issuing play commands and actual media playout can be improved by using Real Time Protocol (RTP) [115] / Real Time Streaming Protocol (RTSP) [116] for media delivery and by negotiating appropriate QoS settings for the signaling of UPnP actions.

In addition, Instead of an http-URL encoded in the bar code a sip-URI could be used instead to take advantage of the remote service discovery and control protocols.

5.6 Portable IMS Gateway in an ad-hoc environment

5.6.1 Objective

Building on the previous feasibility study on media delivery controlled by the mobile phone this study takes that use case further and considers mobile phones in ad-hoc environments, such as inside a car or a boat. This is depicted in Fig. V-14.

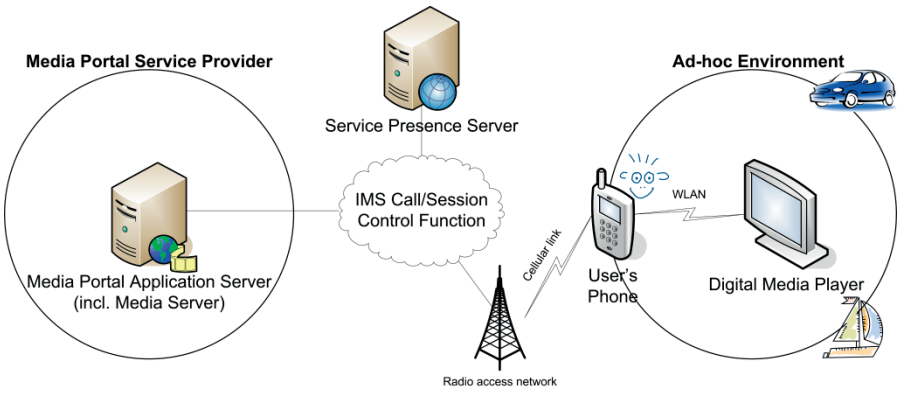


Fig. V-14 Mobile phone in an ad-hoc environment.

In particular, this study looks at how mobile phones equipped with WLAN can establish a bridge between the ad-hoc environment and the Internet over a cellular link. An additional objective is to investigate whether it is feasible that a media provider delivers content to devices within this ad-hoc environment controlled by the mobile phone.

5.6.2 Prototype description

To a large extent existing components from earlier projects were reused in this project. Therefore, this section focuses on only the PIGA component. The internal components of PIGA are shown in Fig. V-15 and described in the following along with how they operate.

A HTC Shift ultra-mobile PC (UMPC) was used for PIGA instead of a normal mobile phone. The advantage is that it allows a standard desktop Linux operating system to be used, which makes prototyping

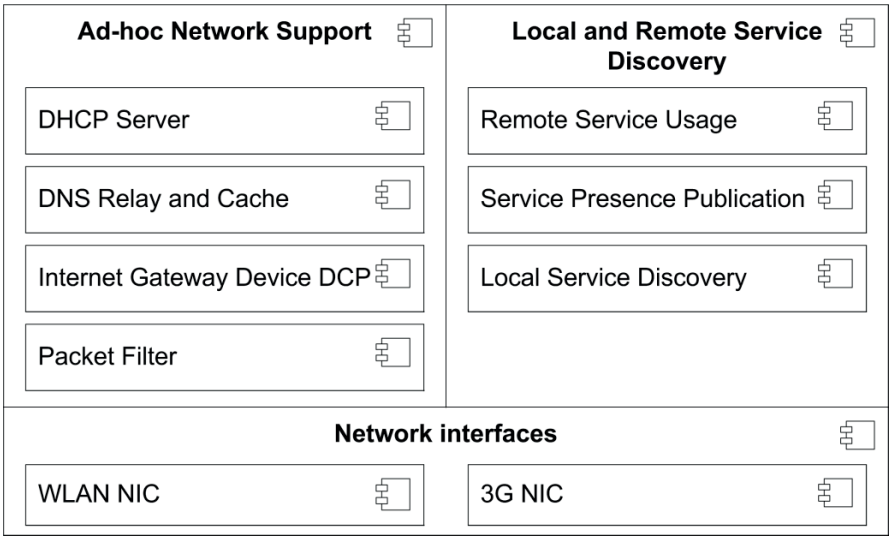


Fig. V-15 Structural view of the Portable IMS Gateway prototype.

quicker and easier due to the breadth of existing functionality that can be used. In particular this

At the bottom of the stack are the two network interfaces that PIGA depends on for connectivity. A WLAN is established using the WLAN network interface card (NIC). Depending on the capabilities of the NIC either an infrastructure mode or an ad-hoc mode network can be established. In the following this network is referred to as ad-hoc even though infrastructure mode could be used. It was found that the NIC embedded in the HTC Shift only supports ad-hoc mode.

To provision network clients with configuration settings Dnsmasq [117] is used as a Dynamic Host Configuratino Protocol (DHCP) [118] server. Most importantly clients will be provisioned with PIGA's local IP address as gateway and DNS server. In addition, Dnsmasq is used as a DNS relay and cache for the local network.

Routing between the local network and the Internet is realized by netfilter [119], which is a packet filtering framework for Linux. This framework enables packet filtering, handling NAT bindings and other packet mangling. Linux UPnP IGD [120] is used to enable IGD (see chapter 6.3A.2.2), to allow local clients to add NAT rules to the packet filter and retrieve the external IP address.

The Local and Remote Service Discovery component discovers local UPnP devices and publishes presence information to the service presence server. When handling remote service usage requests it uses IGD to add NAT rules to allow access to a device.

(a) (b)

Fig. V-16 Screenshot of (a) the media portal login page and (b) the media portal content selection page.

A web browser is used to access the media portal. Users enter their IM Public User identity (IMPU) to login to the portal, as shown in Fig. V-16(a). With web browsers that implements support for the Generic Bootstrapping Architecture (GBA) [121] authentication the IMPU could be sent as a parameter to the request instead. Thereby the login screen would no longer be necessary.

Similar to above, after being authorized access to the media portal users can select content, as shown in Fig. V-16(b), and which media renderer to be used. Using the user's IMPU the media portal subscribes with the service presence server to receive presence information to generate the media renderer list.

5.6.3 Findings

This study shows how users benefit from their mobile phone being able to connect to or create an ad-hoc environment and publish service presence information about the environment. By sharing the presence information with service providers, users can control media delivery to services in their environment.

An issue was found with the HTC Shift that it does not allow using its internal WLAN NIC and 3G NIC at the same time. Clearly, this is problematic when trying to use the two network interfaces together. This issue was worked around by using a 3G USB dongle instead of the internal one, but is something to be aware of with future devices.

Another problem discovered was that operators may give user terminals IP addresses that are not globally routale. This is a policy problem that operators must solve to enable such use cases. Moreover, many operators also do not permit sharing mobile phones' cellular link with other devices, so-called tethering. One problem with tethering is that consumers more easily use cellular bandwidth using their laptops and similar devices as compared to with only their mobile phones. IMS provides a policy framework that allows operators to permit tethering per session, such as to deliver content to media players. In the long term when new radio access networks with more bandwidth are deployed anti-tethering policies may be completely removed as well.

In late September 2009 this solution was successfully demonstrated as part of the Ericsson Roadshow [122] at the 16th World Congress and

Exhibition on Intelligent Transport Systems and Services (ITS) [123] in Stockholm, Sweden, 2009.

Chapter VI

Conclusions

First generic conclusions are given, followed by conclusions on the evaluation and the feasibility studies. Finally, future development opportunities are outlined.

6.1 Generic conclusions

The main motivation and research question for this thesis was to investigate how ubiquitous computing and next-generation networks can benefit from each other. Service discovery in ubiquitous computing was identified as a key area to investigate because it is crucial in ubiquitous computing to be aware of the environment. This research was carried out as part of the Ericsson-University of Agder (UiA) joint-collaboration project ”ONE”. By developing use cases and implementing prototypes we proved that the proposed solutions work.

Normally service discovery protocols support actively sending out search requests for services and passively receive service announcements. By extending the Presence service to support services presence and use it as a remote service discovery protocol, both of these two core service discovery concepts can be supported thanks to the rich semantics the SIP SUBSCRIBE method supports. Moreover, extending the Presence service avoids introducing a new enabler to

NGN. The author believes that this will reduce operating costs for operators. In addition, it makes it easier for developers that can use the same APIs to develop applications for people and services. The patent ”A Method and Apparatus for Service Discovery” [VII] covers these mechanisms, which shows the novelty and innovation in these.

Notably the Presence service extensions for service presence are following the extensibility interfaces of the Presence service, which the author therefore believes will simplify future standardization.

However, because this remote service discovery and control solution uses SIP Presence existing client applications are not supported. For example, current UPnP media players such as those integrated in TVs. In addition, software that mix local services with remote services directly become more complex due to it has to interoperate two different protocol stacks at the same time. Moreover, if two or more control points in the same network use the same remote services, the remote service will send the same event messages to each of the control points. To address these concerns virtualization of remote services was researched, which makes remote services appear like local services. The virtualization function is suitable to be deployed at gateways, including residential gateways and portable gateways in mobile phones. For the remote services, it will appear as one single control point and therefore only single event messages will be sent as well.

Moreover, service providers’ role and how they can leverage remote service discovery and control has been investigated. Use cases for service providers were developed, including media portals, home

security, facility management and health care. In particular, media providers were considered and solution designs for extending existing services and new services were proposed. One new role for media providers is to help customers who cannot directly connect to their environment.

6.2 Evaluation and Feasibility Studies

Several prototypes have been developed as part of this research. A J2ME CDC library for remote access software, named xIGA, was developed that includes core functionality for such software. It includes a UPnP stack for controlling and hosting devices, HTTP client and server stack (mainly to support the UPnP stack), remote UPnP service discovery and control functionality and virtualization of remote services. This library was used to develop a GUI-based application suitable to run on mobile phones and a web-based application suitable for residential gateways. Furthermore, a simple SIP Presence server that supports service presence was also developed. In addition, various support software was developed as well, including a UPnP Low Power Proxy application and a UPnP Media Renderer for Windows.

In addition, a demonstrator for service providers has been developed that includes a media portal (web application) and a hotel room environment. This showcase demonstrates how hotel guests can connect to their hotel room and use services from their service providers at the available appliances.

A problem that we faced during testing of the remote service control solution is that network operators block incoming requests to the IP

addresses provisioned through its Gateway GPRS Support Node (GGSN). For example, to avoid their users to setup HTTP servers. However, this also blocks several use cases for remote service usage. Therefore it should be investigated how network operator policies can be changed to allow remote service-related traffic, while still block unsolicited traffic.

6.3 Future development

Presence scalability is a well-known problem [124-126], and by depending on presence this solution also is affected by this scalability problem. Although the prototypes developed have confirmed that the solutions outlined in this thesis works, it is important that scalability for presence is studied in more detail.

Another area for future study is interoperability between different service discovery protocols, such as UPnP and ZeroConfig. Similar to virtualization of remote services, remote services of a different service discovery protocol may be virtualized as well. A new function for transforming between the service discovery protocols (original and the protocol supported by the target network) must be introduced. This transformation function could be deployed at the same host where the service discovery gateway is deployed or as a network service.

Bibliography

[1] K. Sangani, "Home automation - It's no place like home," *Engineering & Technology,* vol. 1, pp. 46-48, 2006.

[2] M. Weiser, "The computer for the 21st century," *Scientific American,* vol. 265, pp. 66-75, 1991.

[3] M. Satyanarayanan, "Pervasive computing: vision and challenges," *Personal Communications, IEEE [see also IEEE Wireless Communications],* vol. 8, pp. 10-17, 2001.

[4] P. Werle, F. Kilander, M. Jonsson, P. Lönnqvist, and C. Jansson, "A Ubiquitous Service Environment with Active Documents for Teamwork Support," in *Ubicomp 2001: Ubiquitous Computing,* pp. 139-155, 2001.

[5] "UPnP Device Architecture," UPnP Forum, 2003. http://www.upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0.pdf

[6] M. Jeronimo and J. Weast, *UPnP design by example : a software developer's guide to universal plug and play*. Hillsboro, Or.: Intel Press, 2003.

[7] B. A. Miller, T. Nixon, C. Tai, and M. D. Wood, "Home networking with Universal Plug and Play," *Communications Magazine, IEEE,* vol. 39, pp. 104-109, 2001.

[8] Bluetooth, "Specification of the Bluetooth System," 2nd ed, 2004. http://www.bluetooth.com/NR/rdonlyres/1F6469BA-6AE7-42B6-B5A1-65148B9DB238/840/Core_v210_EDR.zip

[9] J. C. Haartsen, "The Bluetooth radio system," *Personal Communications, IEEE [see also IEEE Wireless Communications],* vol. 7, pp. 28-36, 2000.

[10] H. Jaap, N. Mahmoud, I. Jon, J. J. Olaf, and A. Warren, "Bluetooth: vision, goals, and architecture," *SIGMOBILE Mob. Comput. Commun. Rev.,* vol. 2, pp. 38-45, 1998.

[11] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2," RFC 2608, 1999.

[12] E. Guttman, "Autoconfiguration for IP networking: enabling local communication," *IEEE Internet Computing,* vol. 5, pp. 81-86, 2001.

[13] J. E. Katz and S. Sugiyama, "Mobile phones as fashion statements: The co-creation of mobile communication's public meaning," *Mobile communications: Re-negotiation of the social sphere,* pp. 63-81, 2005.

[14] S. P. Walsh, K. M. White, and R. M. Young, "Young and connected: Psychological influences of mobile phone use amongst Australian youth," *Mobile Media 2007,* Sydney, Australia, pp. 125-134, 2007.

[15] H. Ekstrom, A. Furuskar, J. Karlsson, M. Meyer, S. Parkvall, J. Torsner, and M. Wahlqvist, "Technical solutions for the 3G long-term evolution," *Communications Magazine, IEEE,* vol. 44, pp. 38-45, 2006.

[16] International Telecommunication Union, "Recommendation Y.2011: General principles and general reference model for Next Generation Networks," Geneva, Switzerland: Telecommuication Standardization Sector (ITU-T).

[17] G. Camarillo and M. A. García-Martín, *The 3G IP multimedia subsystem (IMS) : merging the Internet and the cellular worlds*, 2nd ed. Chichester: Wiley, 2006.

[18] 3GPP, "IP Multimedia Subsystem (IMS); Stage 2", TS23.228

[19] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, 2002.

[20] P. Belimpasakis, "Seamless User-Generated Content Sharing in the Extended Home," Doctoral dissertation, Faculty of Computing and Electrical Engineering, Tampere University of Technology, 2009. http://URN.fi/URN:NBN:fi:tty-200905271058

[21] F. T. H. den Hartog, M. Balm, C. M. de Jong, and J. J. B. Kwaaitaai, "Convergence of residential gateway technology," *Communications Magazine, IEEE,* vol. 42, pp. 138-143, 2004.

[22] Y. Royon and S. Frenot, "Multiservice home gateways: business model, execution environment, management infrastructure," *Communications Magazine, IEEE,* vol. 45, pp. 122-128, 2007.

[23] K. Hofrichter, "The residential gateway as service platform," *Consumer Electronics, 2001. ICCE. International Conference on,* pp. 304-305, 2001.

[24] B. Horowitz, N. Magnusson, and N. Klack, "Telia's service delivery solution for the home," *Communications Magazine, IEEE,* vol. 40, pp. 120-125, 2002.

[25] M. Ibanez, N. M. Madrid, and R. Seepold, "Virtualization of Residential Gateways," *Intelligent Solutions in Embedded Systems, 2007 Fifth Workshop on,* pp. 115-125, 2007.

[26] "Home Gateway Initiative," accessed Oct. 2009. http://www.homegatewayinitiative.org/

[27] J. Bernstein and T. Spets, "CPE WAN Management Protocol," in *TR-069*: DSL Forum, 2004.

[28] A. Delphinanto, B. A. G. Hillen, I. Passchier, B. H. A. van Schoonhoven, and F. T. H. den Hartog, "Remote Discovery and Management of End-User Devices in Heterogeneous Private Networks," *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pp. 1-5, 2009.

[29] A. E. Nikolaidis, S. S. Papastefanos, G. I. Stassinopoulos, M. P. K. Drakos, and G. A. Doumenis, "Automating remote configuration mechanisms for home devices," *Consumer Electronics, IEEE Transactions on,* vol. 52, pp. 407-413, 2006.

[30] Google, "Google Maps," Aug 2008. http://maps.google.com/

[31] R. Marin-Perianu, P. H. Hartel, and J. Scholten, "A classification of service discovery protocols," Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, Technical report TR-CTIT-05-25, 2005.

[32] F. Zhu, M. W. Mutka, and L. M. Ni, "Service discovery in pervasive computing environments," *Pervasive Computing, IEEE,* vol. 4, pp. 81-90, 2005.

[33] Bluetooth SIG, "Product Directory," accessed Aug. 2008. http://www.bluetooth.com/Bluetooth/Products/Products/

[34] Digital Living Network Alliance, "DLNA Surpasses 5,000 DLNA Certified® Devices Worldwide," Beaverton, Oregon, USA: McGrath/Power Public Relations, 2009. http://www.dlna.org/news/pr/view?item_key=d4925fb999858dad1995c680fe22ef29d4fb3438

[35] UPnP Implementors Corporation, "UPnP Specifications Approved as International Standards," accessed Sept. 2007. http://www.upnp-ic.org/news/pressreleases/UIC_News_Release_on_PAS_ballot_results_Sept07_FINAL.pdf

[36] J. Postel, "Transmission Control Protocol," RFC 793, 1981.

[37] J. Postel, "Internet Protocol," RFC 791, 1981.

[38] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC 2616, 1999.

[39] D. Box, D. Ehnebuske, G. Kakivaya, A. Layma, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Simple Object Access Protocol (SOAP) 1.1," World Wide Web Consortium (W3C), 2000. http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[40] UPnP Forum, "UPnP(TM) Technical Kits," accessed Aug. 2008. http://www.upnp.org/resources/sdks.asp

[41] K. Knightson, N. Morita, and T. Towle, "NGN architecture: generic principles, functional architecture, and implementation," *Communications Magazine, IEEE,* vol. 43, pp. 49-56, 2005.

[42] L. Chae-Sub and D. Knight, "Realization of the next-generation network," *Communications Magazine, IEEE,* vol. 43, pp. 34-41, 2005.

[43] M. Carugi, B. Hirschman, and A. Narita, "Introduction to the ITU-T NGN focus group release 1: target environment, services, and capabilities," *Communications Magazine, IEEE,* vol. 43, pp. 42-48, 2005.

[44] International Telecommunication Union, "Recommendation I.120: Integrated services digital networks (ISDNs)," Geneva, Switzerland: Telecommuication Standardization Sector (ITU-T).

[45] R. Kumar, A. Häber, F. Reichert, and A. Yazidi, "Towards a Relation Oriented Service Architecture," Intelligent Networks: Adaptation, Communication & Reconfiguration (IAMCOM 2010), Fourth Workshop on, Bangalore, India, 2010. (Accepted)

[46] D. C. David, W. John, R. S. Karen, and B. Robert, "Tussle in cyberspace: defining tomorrow's internet," *IEEE/ACM Trans. Netw.,* vol. 13, pp. 462-475, 2005.

[47] L. Deguang, F. Xiaoming, and H. Dieter, "A review of mobility support paradigms for the internet," *Communications Surveys & Tutorials, IEEE,* vol. 8, pp. 38-51, 2006.

[48] International Telecommunication Union, "Recommendation Y.2001: General overview of NGN," Geneva, Switzerland: Telecommuication Standardization Sector (ITU-T).

[49] "Utilization of IMS capabilities Architecture," Open Mobile Alliance, 2005.

http://www.openmobilealliance.org/release_program/ims_v1_0.html

[50] M. Mehdi, No, and C. l, "How IMS enables converged services for cable and 3G technologies: a survey," *EURASIP J. Wirel. Commun. Netw.*, vol. 2008, pp. 1-14, 2008.

[51] ETSI, "Telecoms & Internet converged Services & Protocols for Advanced Network (TISPAN)," accessed Apr. 2007. http://www.etsi.org/tispan/

[52] International Telecommunication Union, "Recommendation Y.2021: IMS for Next Generation Networks," Geneva, Switzerland: Telecommuication Standardization Sector (ITU-T).

[53] K. Rosenbrock, R. Sanmugam, S. Bradner, and J. Klensin, "3GPP-IETF Standardization Collaboration," RFC 3113, 2001.

[54] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol," RFC 3588, 2003.

[55] 3GPP, "IP Multimedia (IM) session handling; IM call model; Stage 2", TS23.218

[56] H. Khartabil, E. Leppanen, M. Lonnfors, and J. Costa-Requena, "An Extensible Markup Language (XML)-Based Format for Event Notification Filtering," 2006.

[57] H. Khartabil, E. Leppanen, M. Lonnfors, and J. Costa-Requena, "Functional Description of Event Notification Filtering," RFC 4660, 2006.

[58] S. Boag, D. Chamberlin, M. F. Fernandez, D. Florescu, J. Robie, and J. Simeon, "XQuery 1.0: An XML Query Language," *W3C Working Draft*, vol. 15, 2002.

[59] 3GPP, "Technical realization of the Short Message Service (SMS)", TS23.040

[60] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, and J. Peterson, "Presence Information Data Format (PIDF)," RFC 3863, 2004.

[61] H. Schulzrinne, V. Gurbani, P. Kyzivat, and J. Rosenberg, "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)," RFC 4480, 2006.

[62] J. Peterson, "A Presence-based GEOPRIV Location Object Format," RFC 4119, 2005.

[63] F. Reichert, A. Häber, M. Gerdes, A. Fasbender, and G. Loudon, ""Sven and the Media Portal" - A Nomadic Use Case for the

Extended Home," *15th IST Mobile & Wireless Communication Summit* Myconos, Greece, 2006.

[64] M. Sujeet, C. Umesh, and D. D. C. Igor, "Movable-multimedia: session mobility in ubiquitous computing ecosystem," *5th International conference on Mobile and ubiquitous multimedia,* Stanford, California: ACM, pp. 2006.

[65] S. Krishnamurthy, O. Anson, L. Sapir, C. Glezer, M. Rois, I. Shub, and K. Schloeder, "Automation of Facility Management Processes Using Machine-to-Machine Technologies," in *The Internet of Things,* pp. 68-86, 2008.

[66] L. Bass, P. Clements, and R. Kazman, *Software architecture in practice*, 2nd ed. Boston, MA, USA: Addison-Wesley Professional, 2003.

[67] "IEEE standard glossary of software engineering terminology," *IEEE Std 610.12-1990,* 1990.

[68] S. J. Vaughan-Nichols, "Presence technology: more than just instant messaging," *Computer,* vol. 36, pp. 11-13, 2003.

[69] Sun Microsystems, "Real-Time, Presence-Based Applications," white paper, 2003. http://www.sun.com/software/products/instant_messaging/im_presence.pdf

[70] M. Debbabi and M. Rahman, "The war of presence and instant messaging: right protocols and APIs," *Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE,* pp. 341-346, 2004.

[71] C. Badulescu, N. Greene, Å. Gustafsson, C. Jaramillo, M. Leclerc, P. Postmus, G. Saavedra, and M. Servant, "Delivering the optimal end-user experience: Ericsson multimedia communication suite," *Ericsson Review,* vol. 2, 2008.

[72] Open Mobile Alliance, "OMA Converged Address Book," 1.0 ed, 2009. http://www.openmobilealliance.org/Technical/release_program/cab_v1_0.aspx

[73] J. Rosenberg, "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)," RFC 4825, 2007.

[74] M. Isomaki and E. Leppanen, "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Manipulating Presence Document Contents," RFC 4827, 2007.

[75] A. Niemi, "Session Initiation Protocol (SIP) Extension for Event State Publication," RFC 3903, 2004.

[76] M. Lonnfors, E. Leppanen, H. Khartabil, and J. Urpalainen, "Presence Information Data Format (PIDF) Extension for Partial Presence," RFC 5262, 2008.
[77] J. Rosenberg, "Presence Authorization Rules," RFC 5025, 2007
[78] J. Rosenberg, "A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)," RFC 3857, 2004.
[79] J. Rosenberg, "An Extensible Markup Language (XML) Based Format for Watcher Information," RFC 3858, 2004.
[80] J. Rosenberg and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)," RFC 3264, 2002.
[81] M. Handley, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol," RFC 4566, 2006.
[82] D. Yon and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)," RFC 4145, 2005.
[83] J. Rosenberg, "The Session Initiation Protocol (SIP) UPDATE Method," RFC 3311, 2002.
[84] P. Belimpasakis and V. Stirbu, "Remote Access to Universal Plug and Play (UPnP) Devices Utilizing the Atom Publishing Protocol," *Networking and Services, 2007. ICNS. Third International Conference on,* pp. 59-59, 2007.
[85] J. Gregorio and B. d. hOra, "The Atom Publishing Protocol," RFC 5023, 2007.
[86] M. Nottingham and R. Sayre, "The Atom Syndication Format," RFC 4287, 2005.
[87] P. Belimpasakis, A. Saaranen, and R. Walsh, "Home DNS: Experiences with Seamless Remote Access to Home Services," *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a,* pp. 1-8, 2007.
[88] A. Häber, F. Reichert, and A. Fasbender, "UPnP Control Point for Mobile Phones in Residential Networks," *15th IST Mobile & Wireless Communication Summit* Myconos, Myconos, Greece, 2006.
[89] JCP expert group JSR-185, "Java(TM) Technology for the Wireless Industry (JTWI)," 2006. http://jcp.org/en/jsr/detail?id=185
[90] JCP expert group JSR-139, "Connected Limited Device Configuration," 1.1 ed, 2003. http://jcp.org/en/jsr/detail?id=139
[91] JCP expert group JSR-118, "Mobile Information Device Profile 2.0," 2002. http://jcp.org/en/jsr/detail?id=118

[92] S. Konno, "CyberLink Development Package for UPnP Devices for Java," accessed Sept. 2009. http://cgupnpjava.sourceforge.net/

[93] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," RFC 3376, 2002.

[94] A. Virolainen and M. Saaranen, "Networked Power Management for Home Multimedia," *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE,* pp. 331-332, 2008.

[95] UPnP Forum, "Low Power Proxy," 1st ed: UPnP Forum, 2007. http://www.upnp.org/specs/lp/UPnP-lp-LowPowerProxy-v1-Service-SDCP-20070828.pdf

[96] C. E. Ortiz, "The Generic Connection Framework," Sun Microsystems Inc., 2003. http://developers.sun.com/mobility/midp/articles/genericframework/

[97] K. Fiveash, "Tao Group throws in the towel," in *The Register*, 2007. http://www.theregister.co.uk/2007/06/13/tao_group_administration/

[98] JCP expert group JSR-172, "J2ME(TM) Web Services Specification," 2004. http://jcp.org/en/jsr/detail?id=172

[99] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL)," World Wide Web Consortium (W3C) 2001. http://www.w3.org/TR/wsdl

[100] "Namespaces in XML," 1st ed, T. Bray, D. Hollander, and A. Layman, Eds. World Wide Web Consortium (W3C) recommendation, 1999.

[101] X. Zheng and F. Chen, "A system architecture for SIP/IMS-based fixed/mobile multimedia services on thin clients," Master's thesis, Faculty of Engineering and Science, University of Agder, 2007. http://student.grm.hia.no/master/ikt07/ikt590/g06/

[102] JCP expert group JSR-218, "Connected Device Configuration (CDC) 1.1," 2005. http://jcp.org/en/jsr/detail?id=218

[103] JCP expert group JSR-216, "Personal Profile 1.1," 2006. http://jcp.org/en/jsr/detail?id=216

[104] JCP expert group JSR-32, "JSIP API Specification," 2003. http://jcp.org/en/jsr/detail?id=32

[105] R. R. Bhat and R. Gupta, "JAIN protocol APIs," *Communications Magazine, IEEE,* vol. 38, pp. 100-107, 2000.
[106] NIST, "Projet IP telephony / VOIP," accessed Mar. 2008. http://snad.ncsl.nist.gov/proj/iptel/nist-sip-downloads.html
[107] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, 1999.
[108] JCP expert group JSR-281, "IMS Services API," 2007. http://jcp.org/en/jsr/detail?id=281.
[109] JCP expert group JSR-116, "SIP Servlet API Specification," 2002. http://jcp.org/en/jsr/detail?id=116
[110] java.net, "Sailfin: SIP Servlet Container," accessed Jan. 2008. https://sailfin.dev.java.net/
[111] P. Kessler, "Ericsson IMS Client Platform," *Ericsson Review,* vol. 2, pp. 50-59, 2007.
[112] J. R. d. M. Gómez, "Service virtualization: bringing a remote personal video recorder into a local network," Master's thesis, Faculty of Engineering and Science, University of Agder, 2009.
[113] E. Gamma, *Design patterns: elements of reusable object-oriented software*. Reading, Mass.: Addison-Wesley, 1994.
[114] ISO/IEC, "Information technology: Automatic identification and data capture techniques – QR Code barcode symbology specification," in *ISO/IEC 18004:2006*, 2006.
[115] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, 2003.
[116] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326, 1998.
[117] S. Kelley, "Dnsmasq - a DNS forwarder for NAT firewalls," accessed Oct. 2009. http://www.thekelleys.org.uk/dnsmasq/doc.html
[118] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, 1997.
[119] "The netfilter/iptables project," accessed Oct. 2009. http://www.netfilter.org/
[120] "The Linux UPnP Internet Gateway Device," accessed Oct. 2009. http://linux-igd.sourceforge.net/index.php
[121] 3GPP, "Generic Authentication Architecture (GAA); Generic bootstrapping architecture", TS 33.220
[122] Ericsson, "Ericsson Roadshow," accessed Oct. 2009. http://www.ericsson.com/campaign/roadshow/

[123] "16th World Congress and Exhibition on Intelligent Transport Systems and Services (ITS)," accessed Oct 2009. http://www.itsworldcongress.com/

[124] P. Bellavista, A. Corradi, and L. Foschini, "IMS-based presence service with enhanced scalability and guaranteed QoS for interdomain enterprise mobility," *Wireless Communications, IEEE,* vol. 16, pp. 16-23, 2009.

[125] A. Houri, E. Aoki, S. Parameswar, V. Singh, and H. Schulzrinne, "Presence Interdomain Scaling Analysis for SIP/SIMPLE," IETF Internet draft, work in progress, 2009.

[126] A. Houri, S. Parameswar, E. Aoki, V. Singh, and H. Schulzrinne, "Scaling Requirements for Presence in SIP/SIMPLE," IETF Internet draft, work in progress, 2009.

[127] S. Cheshire, B. Aboba, and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses," RFC 3927, 2005.

[128] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible Markup Language (XML) 1.0," 4th ed, 2006. http://www.w3.org/TR/2006/REC-xml-20060816/

[129] T. B. Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," RFC 3986, 2005.

[130] UPnP Forum, "UPnP AV Architecture," 1st ed, 2007. http://www.upnp.org/specs/av/UPnP-av-AVArchitecture-v1-20020622.pdf

[131] R. Price, C. Bormann, J. Christoffersson, H. Hannu, Z. Liu, and J. Rosenberg, "Signaling Compression (SigComp)," RFC 3320, 2003.

[132] C. Frankston and H. S. Thompson, "XML-Data reduced," Draft 0.21 ed, 1998. http://www.ltg.ed.ac.uk/~ht/XMLData-Reduced.htm

[133] D. C. Fallside and P. Walmsley, "XML Schema Part 0: Primer," in *World Wide Web Consortium (W3C) recommendation*, 2004. http://www.w3.org/TR/xmlschema-0/

Appendixes

Appendix A Further details on UPnP

A.1 UPnP Device Architecture

The UPnP Device Architecture [5] leverages web technologies and introduces extensions to these. An overview of the layered architecture is given in Fig. A-1.

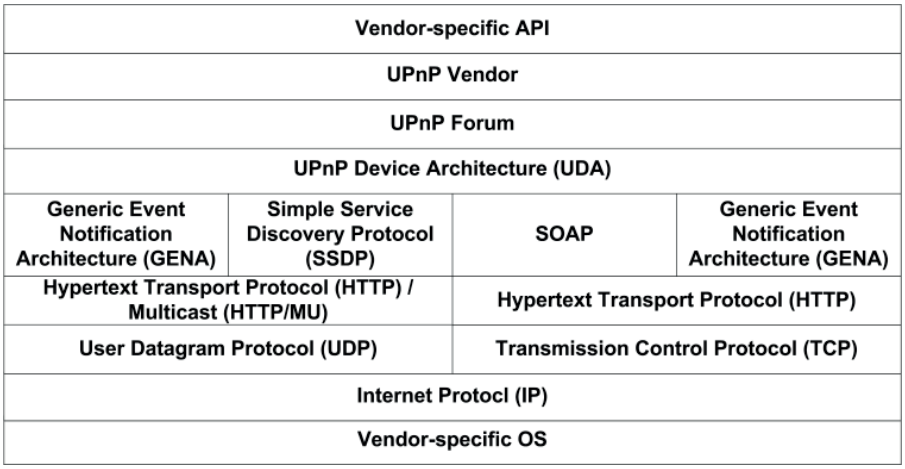


Fig. A-1 Layered UPnP architecture.

The architecture covers six phases that are described in the following with references to the protocols shown in Fig. A-1.

Phase 0, Addressing: All entities must be configured with an IP address for the same local network, either managed by a DHCP server or unmanaged, such as using the Auto-IP protocol [127].

Phase 1, Discovery: Using the Simple Service Discovery Protocol (SSDP) control points discover devices. It supports both query-based (i.e., SEARCH-requests) and announcement-based (i.e., NOTIFY-

requests) discovery. SSDP works in a peer-to-peer fashion with no directory support. The multicast address 239.255.255.250 and the port number 1900, for both UDP and TCP, have been assigned to SSDP by the Internet Assigned Numbers Authority (IANA).

The SEARCH-requests are multicast using HTTP over multicast (HTTP/MU) and are replied to with HTTP over unicast (HTTP/U) from devices that match the search target. In the announcement-based service discovery mechanism the NOTIFY-requests are multicasted, using HTTP/MU, to all peers. NOTIFY-requests should be sent both to advertise that a device is available and when it becomes unavailable.

Device lifetime is an important feature of SSDP. All SEARCH-replies and NOTIFY-requests must include how long the device is available. When the duration is exceeded control points can assume that the device is no longer unavailable, unless a new NOTIFY-request is received in the meantime that extends the device lifetime. Therefore, control points will know that a device is unavilable even if the device goes offline without sending a NOTIFY-request that announces this status change.

Phase 2, Description: The information provided in the discovery phase is very limited. It more or less gives just enough information for a control point to know that the device exists. However, a reference to a device description document is included that has further information about it. This document describes the root device and all embedded devices.

Each device description includes information that is intended for the end-user, such as a friendly name, icons and manufacturer information. In addition, it contains information for the control point, including the device type, the Unique Device Name (UDN) and, optionally, a URL for the presentation phase (described below). In addition, for each device's services the service type, identifier and a reference to an associated service description document is given. Both description document types are encoded as XML [128].

A service description document contains information about the state variables and actions of the service. State variable information includes type and whether it is evented (described below). The actions are related to state variables and used in the Control phase (described below). However, no semantics are included in the description documents. Semantics are only given in the DCPs (described below) by the UPnP Forum.

Phase 3, Control: Services are controlled using the SOAP protocol. A service action is invoked by sending a HTTP POST request to the control URL, which is found in the device description.

Phase 4, Eventing: For eventing the Generic Event and Notification Architecture (GENA) is used, which is also layered on top of HTTP. It extends HTTP with two new request methods: SUBSCRIBE and NOTIFY.

The SUBSCRIBE request is used by control points to subscribe to events from a service of device. When an evented state variable changes the service will send a NOTIFY request to all subscribers with

information about the state change in the message body. This message body is encoded in XML.

Similar to device lifetimes, the event notification subscriptions are also only temporary. Before the subscription time expires the control point should send a new SUBSCRIBE-request to continue to receive event notifications. The SUBSCRIBE-request can also be used to unsubscribe by setting the subscription duration to 0.

Some state variables can change very rapidly, such as the current speed of a fan. For this reason the UPnP Forum standards are augmented with two values, maximumRate and minimumDelta, that describe how often events should be published.

Phase 5, Presentation: Optionally, a device can include web-pages that gives information and allows to control it. In that case a Uniform Resource Locator (URL) [129] for the presentation is included in the device description. The presentation URL can for example be opened with a standard web browser.

A.2 Device Control Protocol

The UPnP Forum standardizes different device types, including service types, in so-called DCPs. A DCP generally includes an architecture description that includes how the device types interwork with each other and specifications for the device- and service types included. A device type specification specifies which devices it embeds and services it offeres. Similary, a service type specification specifies the actions and state variables it has.

In the following, DCPs for audio/video and networking are briefly described.

A.2.1 Audio / Video DCPs

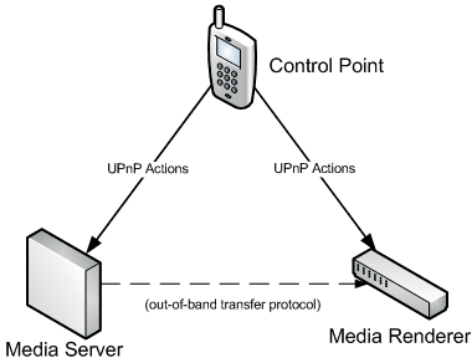


Fig. A-2 UPnP AV Device Interaction Model.

The four audio / video DCPs, Media Server and Media Renderer version 1 and 2, are part of the UPnP AV Architecture [130]. Version 2 is relatively new and of the total number of certified AV products only two of them are Version 2 products. Therefore, only Version 1 will be covered here.

As illustrated in Fig. A-2, UPnP AV works with a control point that discovers media server and media renderer devices. Since these are logical functions they may be deployed on the same hardware device. Also the control point functionality can be deployed at the same device. For example, a TV could provide all of these three functions: media servers that offer the broadcast channels from available tuners, a media renderer to render media, and user interface to control these. As indicated in the future work of [88], a mobile phone is also an ideal device that includes all these three functionalities.

The media server offers content and let the control point browse and search for media. Optionally, the media server can support that a control point imports content to it as well.

An example interaction between those three entities is given in Fig. A-2. When the control point has content it wants rendered at a media renderer device, it retrives the Universal Resource Identifier (URI) [129] of the selected content item and pass it to the media renderer. Because URIs are used it is possible to refer to multiple transfer protocols, both network protocols, such as HTTP [38], RTSP [116], and internal protocols, such as referring to tracks on Audio CDs at the media server. The transfer protocol depends on what is supported by these two devices, and should be taken of by the control point using the Connection Manager Service of both devices.

When the content is rendered, control points may receive event notifications with status of the rendering. In addition, the rendering may be controlled using the Rendering Service.

A.2.2 Networking DCPs

So far two networking DCPs have been standardized: Internet Gateway and WLAN Access Point. The former device type allows to control NAT [107] mappings, retrieve the external address of the device and other functions. With the latter device type, the WLAN Access Point may be configured, such as the frequency channel being used and security details.

Appendix B Further details on IMS

In Fig. B-3 the IMS reference architecture is shown, and in the following the core components are introdcued

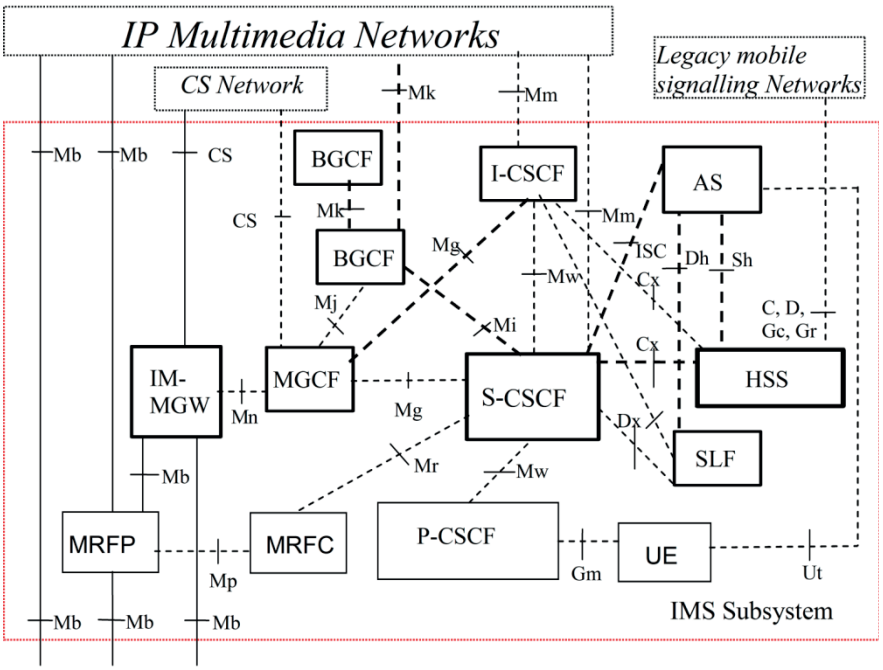


Fig. B-3 Reference Architecture of the IP Multimedia Core Network Subsystem. From Figure 4.0 of [18].

B.1 Application Server (AS)

Services are provided by an application server, via the IP multimedia Subsystem Service Control Interface (ISC), which is based on SIP. Based on the subscriber's service profile the Serving-CSCF selects

application servers to be included, both during session intialization and session termination.

An AS can operate in five different modes. First, an AS can act as a terminating User Agent (UA). A special case is that the AS terminates the request by redirecting it.

Next, the AS can act as the originating UA. One example of this is a "wake up call service" that calls the subscriber at a certain time.

The third operating mode is proxy, where the AS may add, remove or modify the header content of the request it receives and pass it back to the S-CSCF which will proxy it towards the terminating UA.

Most complex is the fourth operating mode, third party control (3PCC). In this mode the AS takes the role of both terminating and originating UA. Examples of 3PCC usage are conferencing and "click-to-dial" services.

Finally, the last operating mode is that the AS is not involved in the session. This includes when the AS is involved only in the session establishment or session termination phase.

B.2 Call Session Control Functions (CSCF)

The CSCFs are responsible for session control. There are four different kinds of CSCF with different roles: Serving-CSCF (S-CSCF), Interrogating-CSCF (I-CSCF), Proxying-CSCF (P-CSCF) and Emergency-CSCF (E-CSCF).

First, the S-CSCF acts as both a SIP registrar by handling registration requests, but also as a proxy when it comes to session control. As

described above, the S-CSCF is responsible for selecting application servers that shall be part of a session. The S-CSCF is *always* located in the subscriber's *home network*.

Next, the I-CSCF is the entry-point to a network. Since there can be many S-CSCFs (and, the other types of CSCF) the I-CSCF is responsible to select an appropriate S-CSCF for a subscriber during registration, based on a set of capabilities. It is usually located in the home network, as well, but it is possible to locate it in the visited network.

The P-CSCF is the user equipment's (UE) entry-point to the IMS. It is always located in the same network as the UE. As the gate between the managed IMS network and the outside, the P-CSCF is responsible for taking out information that should not be exposed to the outside, such as charging information. In addition, it compresses and decompresses the SIP signaling with the UE using signaling compression [131]. This means that the signaling within the core network should always be uncompressed.

Finally, the E-CSCF was introduced in 3GPP Release 7 to support emergency sessions. It is a special CSCF that is alwaays located in the visited network, together with the P-CSCF.

B.3 Home Subscriber Server (HSS)

The HSS is a database of all subscriptions, including their service profile. Furthermore, it handles authentication. The interfaces to the HSS are all based on Diameter.

For scalability, large networks may require several HSS servers. Each subscriber is allocated one HSS server. To locate the HSS handling a

specific subscriber a new functional entity is introduced. This function is named the Subscriber Locator Function.

B.4 Other functional entities

In addition to the abovementioned functional entities, there are several functions for handling media level functionality and interconnecting with other systems, such as Public Switched Telephone Networks (PSTN).

Appendix C
XML Schema definitions for service presence

C.1 XML Schema definition for the UPnP Template Language

The UPnP Device Architecture v1.0 specifies the UPnP Template Language using the XML Data-Reduced (XDR) [132] draft specification. Because PIDF is defined with the W3C XML Schema (XSD) [133] recommendation UPnP Template Language cannot be directly referred to. Therefore the UPnP Template Language was transformed to a XSD version given below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.uia.no/UTL"
  xmlns:utl="http://schemas. uia.no /UTL"
  elementFormDefault="qualified">
  <xs:complexType name="root">
    <xs:sequence>
      <xs:element name="specVersion" type="utl:specVersion"
                  maxOccurs="1" minOccurs="1"/>
      <xs:element name="URLBase" minOccurs="0" maxOccurs="1"
                  type="xs:anyURI"/>
      <xs:element name="device" type="utl:device"
                  maxOccurs="1" minOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="specVersion">
    <xs:attribute name="major" type="xs:int" use="required"/>
    <xs:attribute name="minor" type="xs:int" use="required"/>
  </xs:complexType>
  <xs:complexType name="icon">
    <xs:attribute name="mimetype" type="xs:string"
```

```
                use="required"/>
  <xs:attribute name="width" type="xs:int" use="required"/>
 <xs:attribute name="height" type="xs:int"
                use="required"/>
  <xs:attribute name="depth" type="xs:int" use="required"/>
  <xs:attribute name="url" type="xs:anyURI"
                use="required"/>
</xs:complexType>
<xs:simpleType name="argumentDirection">
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="in"/>
        <xs:enumeration value="xor"/>
        <xs:enumeration value="out"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
<xs:complexType name="action">
  <xs:sequence>
    <xs:element name="argumentList" minOccurs="0"
                maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="argument" maxOccurs="unbounded"
                      minOccurs="1">
            <xs:complexType>
              <xs:attribute name="name" type="xs:string"
                            use="required"/>
              <xs:attribute name="direction"
              type="utl:argumentDirection" use="required"/>
              <xs:attribute name="retval"  use="optional"/>
              <xs:attribute name="relatedStateVariable"
                type="xs:string"  use="required"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"
                use="required"/>
</xs:complexType>
<!-- Should rather make some mapping between the UPnP data
```

```
      types and XML Schema data types
  -->
  <xs:simpleType name="dataType">
    <xs:list>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="ui1"/>
          <xs:enumeration value="ui2"/>
          <xs:enumeration value="ui4"/>
          <xs:enumeration value="i1"/>
          <xs:enumeration value="i2"/>
          <xs:enumeration value="i4"/>
          <xs:enumeration value="int"/>
          <xs:enumeration value="r4"/>
          <xs:enumeration value="r8"/>
          <xs:enumeration value="number"/>
          <xs:enumeration value="fixed.14.4"/>
          <xs:enumeration value="float"/>
          <xs:enumeration value="char"/>
          <xs:enumeration value="string"/>
          <xs:enumeration value="date"/>
          <xs:enumeration value="dateTime"/>
          <xs:enumeration value="dateTime.tz"/>
          <xs:enumeration value="time"/>
          <xs:enumeration value="time.tz"/>
          <xs:enumeration value="boolean"/>
          <xs:enumeration value="bin.base64"/>
          <xs:enumeration value="bin.hex"/>
          <xs:enumeration value="uri"/>
          <xs:enumeration value="uuid"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:list>
  </xs:simpleType>
  <xs:complexType name="stateVariable">
    <xs:choice minOccurs="0" maxOccurs="1">
      <xs:element name="allowedValueList">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="allowedValue" type="xs:string"
                        minOccurs="1" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="allowedValueRange">
```

```
      <xs:complexType>
        <xs:attribute name="minimum" type="xs:float"
                      use="required"/>
        <xs:attribute name="maximum" type="xs:float"
                      use="required"/>
        <xs:attribute name="step" type="xs:float"
                      use="optional"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
  <xs:attribute name="name" type="xs:string"
                use="required"/>
  <xs:attribute name="dataType" type="utl:dataType"
                use="required"/>
  <xs:attribute name="defaultValue" type="xs:string"
                use="optional"/>
</xs:complexType>
<xs:complexType name="service">
  <xs:sequence>
    <xs:element name="scpd" maxOccurs="1" minOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="specVersion"
       type="utl:specVersion" maxOccurs="1" minOccurs="1"/>
          <xs:element name="actionList" minOccurs="0"
                      maxOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="action" type="utl:action"
                  minOccurs="1" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="serviceStateTable"
                      maxOccurs="1" minOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="stateVariable"
                  type="utl:stateVariable" minOccurs="1"
                  maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
```

```
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="serviceType" type="xs:anyURI"
                  use="required"/>
    <xs:attribute name="serviceId" type="xs:anyURI"
                  use="required"/>
    <xs:attribute name="controlURL" type="xs:anyURI"
                  use="required"/>
    <xs:attribute name="eventSubURL" type="xs:anyURI"
                  use="required"/>
  </xs:complexType>
  <xs:complexType name="device">
    <xs:sequence>
      <xs:element name="iconList" minOccurs="0"
                  maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="icon" type="utl:icon"
              minOccurs="1" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="serviceList" maxOccurs="1"
minOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="service" type="utl:service"
              minOccurs="1" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="deviceList" maxOccurs="1"
        minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="device" type="utl:device"
              minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="deviceType" type="xs:anyURI"
      use="required"/>
    <xs:attribute name="friendlyName" type="xs:string"
```

```
      use="required"/>
    <xs:attribute name="manufacturer" type="xs:string"
                  use="required"/>
    <xs:attribute name="manufacturerURL" type="xs:anyURI"
                  use="optional"/>
    <xs:attribute name="modelDescription" type="xs:string"
                  use="optional"/>
    <xs:attribute name="modelName" type="xs:string"
                  use="required"/>
    <xs:attribute name="modelNumber" type="xs:string"
                  use="optional"/>
    <xs:attribute name="modelURL" type="xs:anyURI"
                  use="optional"/>
    <xs:attribute name="serialNumber" type="xs:string"
                  use="optional"/>
    <xs:attribute name="UDN" type="xs:anyURI"
                  use="required"/>
    <xs:attribute name="UPC" type="xs:string"
                  use="optional"/>
  </xs:complexType>
</xs:schema>
```

C.2 XML Schema for service presence

The following schema extends PIDF with the upnp-root-device element to include UPnP device and service descriptions in the document.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:svcpres="http://schemas.uia.no/ServicePresence"
  xmlns:pidf="urn:ietf:params:xml:ns:pidf"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:utl="http://schemas.uia.no/UTL"
  targetNamespace="http://schemas.uia.no/ServicePresence"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="urn:ietf:params:xml:ns:pidf"
    schemaLocation="pidf.xsd"/>
  <xs:import namespace="http://schemas.uia.no/UTL"
    schemaLocation="UTLv2.xsd"/>
  <xs:annotation>
    <xs:documentation>
      Service presence tuple extensions for PIDF.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="upnp-root-device" type="utl:root"/>
</xs:schema>
```

C.3 Example PIDF document with service presence

In the following an example document using the service presence extension is given.

```
<?xml version="1.0" encoding="UTF-8"?>
<presence
  xmlns="urn:ietf:params:xml:ns:pidf"
  entity="pres:sip:higa.home1@ims.ict-fiesta.test">
  <tuple id="uuid:89665984-7466-0019-5b46-051c73783736">
    <status>
      <basic>open</basic>
    </status>
    <upnp-root-device
      xmlns="http://schemas.uia.no/ServicePresence">
      <device
        xmlns="http://schemas.uia.no/UTL"
        UDN="uuid:89665984-7466-0019-5b46-051c73783736"
        deviceType="urn:schemas-upnp-
                    org:device:MediaServer:1"
        friendlyName="Media Server by TwonkyVision"
        manufacturer=""
        modelName="">
        <serviceList>
          <service
controlURL="http://192.168.1.9:9000/ContentDirectory/Control"
eventSubURL="http://192.168.1.9:9000/ContentDirectory/Event"
            serviceId="urn:upnp-
                      org:serviceId:ContentDirectory"
            serviceType="urn:schemas-upnp-
                         org:service:ContentDirectory:1"/>
          <service
controlURL="http://192.168.1.9:9000/ConnectionManager/Control
"
eventSubURL="http://192.168.1.9:9000/ConnectionManager/Event"
            serviceId="urn:upnp-
                      org:serviceId:ConnectionManager"
            serviceType="urn:schemas-upnp-
                         org:service:ConnectionManager:1"/>
        </serviceList>
      </device>
    </upnp-root-device>
  </tuple>
</presence>
```

REMOTE SERVICE USAGE THROUGH SIP WITH MULTIMEDIA ACCESS AS A USE CASE

Andreas Häber
Agder University College
Grimstad, Norway

Martin Gerdes
Ericsson Research
Aachen, Germany

Frank Reichert
Agder University College
Grimstad, Norway

Andreas Fasbender
Ericsson Research
Aachen, Germany

Ram Kumar
Agder University College
Grimstad, Norway

***Abstract*—The IP Multimedia Subsystem is under deployment, as an IP-based service control and access infrastructure, but how it interconnects with residential appliances is currently unclear. With IMS access for the residential appliances they can be used as both service consumers and service providers. In this paper we present a protocol which allows residential services to be remotely invoked, through the IMS, and consumed in a different network, along with a prototype implementation and early results. With our protocol services of two distinct service protocol systems can cooperate.**

***Index Terms*—IP Multimedia Subsystem, multimedia communication, telecommunication services, service islands, wireless LAN, multimedia systems**

I. INTRODUCTION

ERICSSON and Agder University College are looking into residential services in fixed-mobile converged networks. The IP Multimedia Subsystem (IMS) [1] is under deployment as an IP-based service control and access infrastructure. Devices, mobile or fixed, can register to central identification and access control nodes to obtain access to IP based services provided by the IMS infrastructure in a secure manner. IMS offers services through mobile or fixed Wide Area Networks (WAN) supporting Quality of Service (QoS).

In this paper, we start by presenting example use cases for providing and using remote services in Section II. Subsequently, in Section III, a protocol for enabling these use cases is proposed, based on the Session Initiation Protocol (SIP) [2]. We describe a prototype implementation of this protocol in Section IV, along with early results of the prototype in Section V.

In Section V an overview of related work is given, and finally, in Section VII, conclusions from our current results are stated, and we describe future work for both the protocol and the prototype.

II. USE CASES

What if the IMS connected device is in a house with access to a local network? In the local network, many services and resources are available, such as a TV, game console, home automation, etc. To enable a better user experience it is important that they can interconnect with the IMS. The device uses service discovery protocols, such as Service Location Protocol (SLP) [3], Universal Plug & Play (UPnP) [4], Bluetooth, etc, to be aware of the resources in the local network, as shown in Figure 1.

A. IMS Service consumption in a local network

Video service usage, where the consumer routes the content stream to a TV instead of its mobile phone, is an example of using IMS services together with the home network's resources. Consumers can preview movies, free of charge, on a mobile terminal and can pay to receive the content stream in a quality suitable to be viewed full screen on a TV screen. As the TV/Set-top Box (STB) typically will not host an IMS client, a gateway between IMS and the home network is required, referred to as Home IMS Gateway (HIGA) in Figure 1.

If the content stream is available in low quality only, for instance mobile originated video call, the rest of the display area can be used to show data related to the video call. In business scenarios, this can be spreadsheets, presentations, etc, while in family scenarios this can be a picture slideshow, browsing websites together for travel planning, etc.

B. Providing services from the local network

Home networks also provides services, such as access to Heat, Ventilation and Air Conditioning (HVAC) appliances, multimedia, and digital security cameras. With the expected pervasive deployment of IMS, users can therefore access their services and resources from everywhere: at the office, while visiting friends, when traveling, etc.

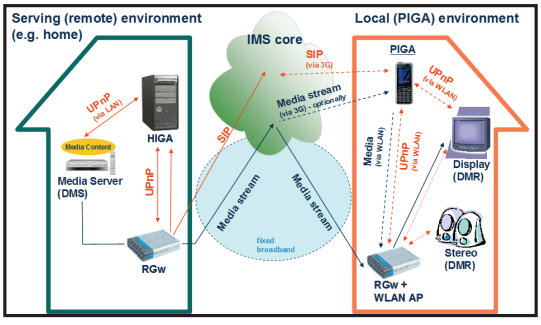


Figure 1 Remote media access use case.

1-4244-1144-0/07/$25.00 ©2007 IEEE.

This allows house owners on tour to not only receive a notification from sensors, which is available today with the second 2G Short Messaging Service (SMS), but also to consume the video stream of the digital security camera by taking advantage of the security and QoS provided by the IMS infrastructure. The identification support of IMS helps devices to authenticate and connect to the home network, independent of current location and access network.

C. *Remote service cooperation*

So far, we have considered a service in the local network to either consume or provide services. However, there are also scenarios where users would like services of two local networks to cooperate. Access to multimedia content stored in the home network is one scenario. The user has a home network that uses any type of service discovery mechanism (e.g., UPnP or SLP), including a media server where the family's multimedia files are stored: pictures, high definition videos, music, etc.

However, when family members leave their home they cannot access the services in their home network, because many service discovery mechanisms, such as UPnP, rely on all services being part of the same local network. This is good for security, because it makes it harder for intruders to access the services.

For example, a family member stays at a hotel while traveling. It is now common to expect a TV in hotel rooms today, and some hotels provide broadband access with Wireless Local Area Network (WLAN) for their guests. The hotel room network also supports a service discovery mechanism, and it must cooperate with the guest's home network to allow it to, for example, consume multimedia, as shown in Figure 1. IMS makes it easy to locate and access the guest's home network in a secure manner, and ensures quality for the content stream.

D. *A gateway function for remote service invocation*

As mentioned above, the service discovery mechanisms inside local networks cannot interoperate between backbone interconnections. For example, UPnP uses multicasting for service discovery but the multicast communication is restricted to the local network, and the service addresses should be from the private address space [5]. Moreover, IMS is not aware of the services provided within the local networks. Therefore, gateway functionality between the local networks and the wide area interconnection is necessary to enable the scenarios outlined above.

III. Remote Service Invocation Protocol

The remote service invocation protocol, which we have named PIRANHA, is based on SIP. It works between two Service Discovery Gateways (SDGs) connecting independent service discovery mechanisms together, through a common backbone. In the media plane, service commands and corresponding content are sent to the services controlled by the terminating SDG. This is different to other solutions, mentioned in Section VI, which use the session plane for service requests.

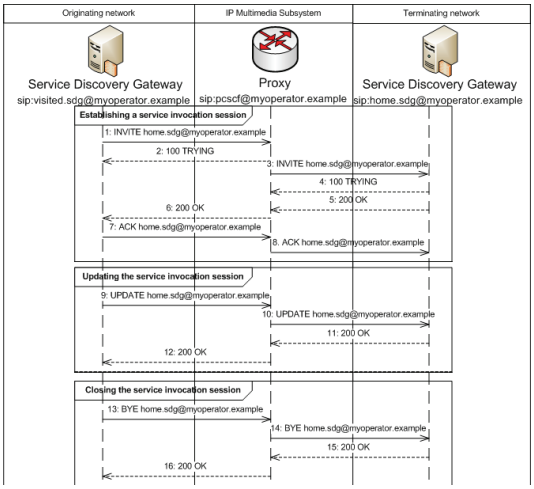


Figure 2 Message sequence at the signal plane for a remote service invocation session.

For remote connectivity scenarios, such as in the "hotel room to home network" scenario described in Section II.C, IMS can provide support for the necessary addressing, access control, and QoS. In other scenarios, for instance translation between two service discovery mechanisms in the same Local Area Network (LAN), intermediate nodes between the SDGs may not be necessary. Two SDGs establish service invocation sessions to be able to pass service invocation requests between them. In the following, we refer to the two SDGs as the originating SDG and the terminating SDG, where the originating SDG initiate the session and can pass requests to the terminating SDG.

There are three stages of the protocol: session establishment (Section III.A), update a session (Section III.B), and finally close a session (Section III.C). Command requests can be sent from the originating SDG to the terminating SDG, through the media plane, after a session has been established.

A. *Establish service invocation sessions*

Initiating a session is accomplished by sending an INVITE-

```
1  v 0
2  o visited.sdg 3380446179 3380446179 IN IP4
192.168.168.31
3  s
4  c IN IP4 /192.168.168.31
5  t 0 0
6  a recvonly
7  m application 9 TCP piranha
8  a udn:uuid:9afb3231 345a 4cd1 b448
8866b79ff91b
9
10 a setup:active
```

Figure 3 SDP offer for remote service invocation session.

request with a Session Description Protocol (SDP) body [6], which describes the session, using the offer/answer model [7]. Figure 2 shows the complete message sequence.

In the INVITE-request, the originating SDG requests a device that it would like to control by specifying the udn-attribute in the SDP-offer, similar to the one shown in Figure 3. The session description part (lines 1 to 6) follows [7], therefore we don't describe it here. There is one media description included, starting at line 7, which specifies that the media type is *application* and the p*iranha* format. The terminating side can use this to see if it understands this protocol or not. If the terminating side does not understand application/piranha then it must return an appropriate response code, such as "488 Not Acceptable Here", and the session establishment will be cancelled. The media description line also specifies that we want to use Transmission Control Protocol (TCP) [8] as the transport protocol on port 9, the discard port, following [9], for service invocation requests. Three media-level value-attributes are also included: *udn (*from Unique Device Name (UDN)) is the device the SDG requests to control, and is a new attribute. The other two attributes specifies usage of the TCP port, and are specified in [9]. *Setup:active* means that the offering SDG will initiate an outgoing connection, and the *connection:new* attribute indicates that this should be a new connection. During the service invocation session this value attribute can be changed to *existing* to indicate that there is an existing connection that should be reused.

If the terminating SDG accepts the INVITE-request, including its offer, it will add a port binding for this service invocation session at an arbitrary port at the local network's gateway, for example using UPnP. The SDP-answer, from the terminating SDG, looks similar to the offer, except that it specifies the destination port for the TCP-connection, and the setup and connection attributes specifies that it will listen for connections (setup:passive) and use a new connection (connection:new). The originating SDG will then receive a 200 OK response.

Now the originating SDG will check the answer in the response, and, if it is acceptable, it will send an ACK-request to the terminating SDG, via the proxy, to confirm it. Then it can start to use the media connection to invoke services. However, if the answer is unacceptable the originating SDG must terminate the session by sending a CANCEL-request instead, or, if supported, it can alter the session by sending an UPDATE-request.

Because the terminating SDG makes a Network Address Translation (NAT) [10, 11] mapping between the service's private internet address and the public internet address, it does not need to handle any of the traffic at the media plane. This makes the design and operation simple, but in some cases too simple because information that the service sends may include, for example, address information specific to its private network, causing problems when used in the originating network. An example of this issue is the "resource-uri" included in browse-results from UPnP Media Servers, which we discuss below.

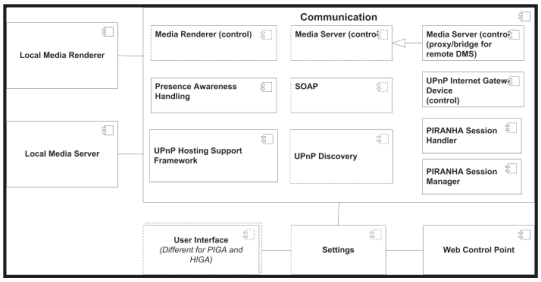


Figure 4 Logical view of xIGA Library, with user interface component for HIGA and PIGA. Dashed lines indicate components from our "Mobile UPnP Control Point" prototype.

B. *Updating service invocation sessions*

During the session it might be necessary to alter the session description, for example to add or remove a media description for a media stream. To update the session an SDG can send a new INVITE-request for the same dialog, or, if supported, an UPDATE-request [12], including a new session description documents with the updated session information. Sequence 9-12 in Figure 2 shows this.

Being able to update the session is especially important when accessing IMS through cellular networks, where resources are scarce and it may be expensive for the customer to occupy resources.

C. *Closing the service invocation session*

Both SDGs can close the session, but normal behavior is that the originating SDG closes it. In exceptional situations, for example if the service becomes unavailable, the terminating side should close it instead.

Sending a BYE-request to the other SDG will close the session, as shown in sequence 13-16 in Figure 2.

All resources associated with the session must be released when the session is closed, including any port mappings setup during the course of events of the session.

IV. PROTOTYPE FOR REMOTE SERVICE INVOCATION

We have implemented a prototype of the establishment phase (Section III.A) by modifying our "Mobile UPnP Control Point" [13]. First, we had to port that prototype from a platform based on Java 2, Micro Edition (J2ME) Connected Limited Device Configuration (CLDC) to J2ME Connected Device Configuration (CDC).

In this section, we first describe why we changed our platform to J2ME CDC, and then we present the architecture of our new prototype.

A. *Motivation for porting to J2ME CDC*

As described in [4] the multicasting support in J2ME CLDC

1-4244-1144-0/07/$25.00 ©2007 IEEE.

is insufficient for UPnP, which was our main reason for switching to J2ME CDC as our platform.

In addition, we found that the reflection support in J2ME CLDC was too limited and made our API design for hosting UPnP devices awkward to use. In retrospect, our design for device hosting might be too powerful for this constrained platform. Our device hosting support allows any device, which uses our APIs, to be registered at runtime, which is why we require reflection. By altering the design, so devices are rather registered statically, we do not depend on so much reflection usage at runtime.

B. Scenario description

The scenario we consider here is users away from their home network, but the users still would like to access their multimedia. Examples of such situations are when visiting friends or staying at a hotel.

Even though mobile devices' storage space grows fast, is it hard to imagine that it will outgrow the pace of storage space required for a family's multimedia collection (family pictures, movies in high-definition format, and lossless music). Therefore, an access mechanism to a remote multimedia server is needed.

C. Deployment of PIRANHA Prototype

In our prototype system, we have three networks: home, visited, and an operator network connecting them together, as depicted in Figure 5. There is a gateway-device at the edge of both the home and visited networks, which implements the UPnP Internet Gateway Device (IDG) Device Configuration Protocol (DCP).

In the home network, the Digital Media Server (DMS) is located with the family's multimedia, whereas the Digital Media Player (DMP) is in the visited network.

D. Logical view of the PIRANHA Prototype Architecture

The prototype software is deployed at two nodes, HIGA[14] and Portable IMS Gateway (PIGA), which share a common library, named xIGA, for UPnP and the remote access functionality. This library is an evolution of the prototype described in [4], and the design is depicted in Figure 4. We will focus on describing the functionality relevant to the protocol described in Section III here.

1) *PIRANHA Session Manager:* This component implements the client part of the protocol. For instance, a user enters the address of the SDG in the user interface, which results in the PIRANHA Session Manager will establish a remote service invocation session with that SDG.

When creating the INVITE request it must also create the SDP-offer and setting the udn-attribute to a value known a priori to exist in the home network. In our testing, we only support UPnP AV Media Servers, and therefore do not use this value.

2) *PIRANHA Session Handler:* Handles requests about session establishment and, in the future, termination of sessions.

It will parse the SDP-offer and see if it can discover the requested device. If the device is available then it adds a port mapping for the device's private IP address at the IGD for the media plane.

3) PIRANHA Session: Common for both the PIRANHA Session Manager and PIRANHA Session Handler, and has information about port mapping and the SIP dialog. When a message comes in (SIP request or response) those components can look up the session information in a table, based on the dialog identifier.

4) *Media Server Proxy:* This component helps to make PIRANHA transparent for the user interface in PIGA. The user interface sees it as a normal media server and use it as the normal Media Server control objects, because the proxy inherits from that class. However, this proxy sends requests to the external DMS, in the media plane of the PIRANHA session.

When browsing the DMS the browse result includes a Uniform Resource Identifier (URI) [15]. Usually this URI contains a private network address, such as 192.168.1.3 in our scenario, because the DMS does not expect requests from external networks. Therefore, the Media Server Proxy must change this element of the browse result so the DMP can locate the resource, if the user requests to play it.

V. Prototype Results

In our initial test configuration, with the home and visited networks one hop away from each other, we have successfully streamed music from a UPnP Media Server, TwonkyVision Media Server, to a UPnP Media Renderer, D-Link DSM320, with HTTP as the transport protocol, as required by DLNA.

As expected, video streaming over HTTP does not work very well over WAN. Our preliminary investigation of this issue indicates that the problem is related to the TCP sliding window mechanism.

With our prototype client, we have shown that it is feasible on a standard mass-market phone to support both the SIP and

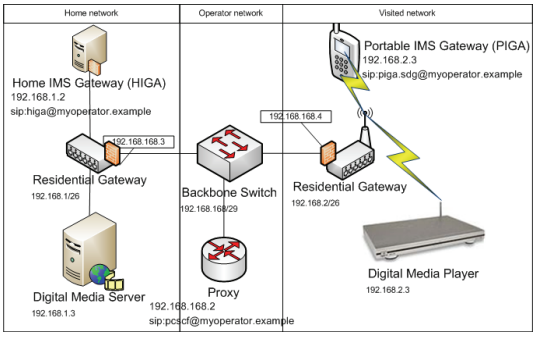


Figure 5 Deployment of the prototype system.

the UPnP stacks simultaneously. This makes it possible to realize new fixed-mobile converged applications.

VI. Related work

In [16] a similar protocol for remote service invocation is proposed, however with a different approach to handling the "resource-uri" issue. They rather let the terminating network's gateway function be part of the media plane, and mediate the information sent between the two networks.

Other solutions exist as well, such as [17, 18], where the service invocation requests are sent using the SIP MESSAGE-requests. In our opinion, it is however better to separate the signaling and the media plane for more flexibility, such as adding more media streams to the session.

VII. Conclusion and future work

Our prototype shows that our protocol works and allows services described by different service discovery mechanisms to cooperate, through the introduced Service Discovery Gateway function. However, HTTP streaming, which is the only mandated transport protocol of the Digital Living Network Alliance (DLNA), is causing issues related to the TCP flow control mechanism that should be further investigated.

We will later investigate how the protocol behaves over a longer distance, with more hops between the home and visited networks, and this might result in further TCP problems.

Our solution for fixing the "resource-uri" issue works, but we will continue to look for other options, which requires less knowledge of a service's semantics. Related to this issue is that our solution is limited by the cascading NAT problem [19], which we will also investigate further.

Finally, our current prototype does not handle updating and closing the session, and does not handle remote service discovery. Session updating and closing is described above, and the mechanism for selecting the service to be controlled is included in the protocol design (the udn-attribute). Remote service discovery is important to provide a good user experience.

Acknowledgment

The authors are grateful for the contributions of Xianghan Zheng, for his work with porting our software from J2ME CLDC to J2ME CDC, and Xiaochun Xu, for his work with implementing and testing the prototype.

References

[1] G. Camarillo and M. A. García-Martín, *The 3G IP multimedia subsystem (IMS) : merging the Internet and the cellular worlds*, 2nd ed. Chichester: Wiley, 2006.

[2] J.Rosenberg, H.Schulzrinne, G.Camarillo, A.Johnston, J.Peterson, R.Sparks, M.Handley, and E. Schooler, *SIP: Session Initiation Protocol*, IETF RFC 3261, http://www.rfc-editor.org/rfc/rfc3261.txt.

[3] E.Guttman, C.Perkins, J.Veizades, and M. Day, *Service Location Protocol, Version 2*, IETF RFC 2608, June, 1999; http://www.rfc-editor.org/rfc/rfc2608.txt.

[4] M. Jeronimo and J. Weast, *UPnP design by example : a software developer's guide to universal plug and play*. Hillsboro, Or.: Intel Press, 2003.

[5] Y.Rekhter, B.Moskowitz, D.Karrenberg, G. J. d. Groot, and E. L. L. rfc, "Address Allocation for Private Internets," 1996.

[6] M.Handley, V.Jacobson, and C. P. L. rfc, "SDP: Session Description Protocol," 2006.

[7] J.Rosenberg and H. S. L. rfc, "An Offer/Answer Model with Session Description Protocol (SDP)," 2002.

[8] J. P. L. rfc, "Transmission Control Protocol," 1981.

[9] D.Yon and G. C. L. rfc, "TCP-Based Media Transport in the Session Description Protocol (SDP)," 2005.

[10] P.Srisuresh and M. Holdrege, *IP Network Address Translator (NAT) Terminology and Considerations*, IETF RFC 2663, http://www.rfc-editor.org/rfc/rfc2663.txt.

[11] P.Srisuresh and K. Egevan, *Traditional IP Network Address Translator (Traditional NAT)*, IETF RFC 3022, http://www.rfc-editor.org/rfc/rfc3022.txt.

[12] J. R. L. rfc, "The Session Initiation Protocol (SIP) UPDATE Method," 2002.

[13] A. Häber, F. Reichert, and A. Fasbender, "UPnP Control Point for Mobile Phones in Residential Networks," in *15th IST Mobile & Wireless Communication Summit*. Myconos, Greece, 2006. http://mobilesummit2006.org/

[14] T. Cagenius, A. Fasbender, and L. Barriga, "An IMS Gateway for Service Convergence in Connected Homes," in *45th Congress of the Federation of Telecommunications Engineers of the European Community (FITCE)*. Athens, Greece, 2006.

[15] T. B. Lee, R.Fielding, and L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, IETF RFC 3986, http://www.rfc-editor.org/rfc/rfc3986.txt.

[16] O. Yeon-Joo, L. Hoon-Ki, K. Jung-Tae, P. Eui-Hyun, and P. Kwang-Roh, "The DLNA Proxy System Architecture for Sharing In-Home Media Contents via Internet," 2006.

[17] B. Kumar and M. Rahman, "Mobility support for universal plug and play (UPnP) devices using session initiation protocol (SIP)," 2006.

[18] A. Brown, M. Kolberg, D. Bushmitch, G. Lomako, and M. Ma, "A SIP-based OSGi device communication service for mobile personal area networks," 2006.

[19] R. J. Vijay K. Gurbani, "Contemplating some open challenges in SIP," *Bell Labs Technical Journal*, vol. 9, pp. 255-269, 2004.

Using SIP Presence for Remote Service Awareness

Andreas Häber[1], Martin Gerdes[2], Frank Reichert[1], Andreas Fasbender[2], Ram Kumar[1]
[1]University of Agder, [2]Ericsson GmbH

Abstract

Residential networks usually protect its devices and services behind firewalls and use private IP addresses. Therefore, appliances within a residential network cannot be discovered and utilized from external networks by standardized technologies as UPnP. In this paper, we present our concept of "Service Presence", based on the 3GPP Presence Service that makes the service presence information remotely discoverable.

Introduction

Residential services are today becoming digitalized. This opens up for new possibilities in the home that are not possible with their analog counterparts. For example, photo albums can be replaced by media servers with digital photos. An important feature with digital residential services is that they can be interconnected and cooperate throughout multiple homes, users and devices.

For services to cooperate, they must become aware of each other and establish relationships. Many service discovery mechanisms already exist that enable this, such as Universal Plug & Play (UPnP), Bluetooth and ZigBee. With these mechanisms, the residential service network follows Service Oriented Architecture (SOA) principles.

However, all these mechanisms have limited coverage. UPnP is limited to private, local networks, whereas Bluetooth and ZigBee depend on the coverage area limited by the physical characteristics of their respective transmission technologies. Therefore, the services cannot be accessed from remote networks. This limitation is good for illegitimate access, but there are also legitimate scenarios that should be allowed. One such scenario is when the user is in a remote network and wants to access his personal digital photo album, which is located on a server in his home network. Another scenario is to allow service providers to deliver content to residential devices, such as delivering IPTV to any media player in the residential network.

Contributions

In [1] use cases for remote service access and a protocol for remote service usage are presented. However, usage of remote services depends on the knowledge of which services are available in the remote network. Within this paper, a solution for that problem is presented and a description of a prototype implementation is given. The solution compliments the Remote Service Usage protocol.

In the Remote Service Usage protocol the Session Initiation Protocol (SIP) [2] is used to establish the connectivity between the remote service and the user. Because SIP is also the core signaling protocol of the IP Multimedia Subsystem (IMS), Remote Service Usage can leverage the additional capabilities provided by IMS, including Quality of Service (QoS) and authentication, authorization and accounting (AAA). To also be able to leverage the IMS for remote service awareness the protocol is based on the 3GPP Presence Service [3]. Therefore, the solution has been termed "Service Presence".

This paper was presented at the NIK 2008 conference. For more information see http://www.nik.no/

Structure of this paper

This paper is structured as following. First, background information is presented in the following section. Next, the "Service Presence" concept is introduced. Following, the protocol for "Service Presence" is described. Then, a prototype implementation of this protocol is shown followed by some early results. Finally, conclusions are given.

Background

Residential Networks and Service Discovery

Service discovery [4-6] is a mechanism to discover, and be aware of, services. Many standards for service discovery protocols exist [7-8], such as UPnP [9-10], Apple Bonjour [11], and Bluetooth [12].

UPnP provides standardized methods to describe and exchange device profiles, including available services (so-called *actions*) provided by the device, and their respective capabilities and requirements. The UPnP Forum standardizes Device Control Packages (DCP) for device types. One of these is the UPnP Audio-Video (AV) Architecture [13]. This architecture describes control points, media servers and media renderers, and relationships between them. In [14] an implementation of such a control point running on mobile phones is shown that can browse and select content from a media server and set a media renderer to play the selected content. The Digital Living Network Alliance (DLNA) architecture [15-16] leverages the UPnP AV architecture. It is used here as the foundation for user-friendly applications that require communication between entertainment devices within private homes through IP networks.

Similarly in Bluetooth, a service discovery protocol (SDP) has been standardized to search and identify other Bluetooth devices and their services in the vicinity.

However, these service discovery mechanisms only work in local networks, as described in [1]. Thus, they do not allow discovering services in an external network, and, consequently, do not support access to remote services. Use cases for such remote service access are described in [1] together with a proposal for a remote service usage protocol.

IP Multimedia Subsystem

IMS [17] is under deployment as IP based service control infrastructure. Devices (mobile or fixed) can register to central identification and access control nodes to obtain access to IP based services provided by the IMS infrastructure.

SIP is the underlying control protocol of IMS to initiate, operate and terminate so-called sessions between service providers and service consumers. In [18] a detailed outline of IMS' service layer is given.

Presence with the Session Initiation Protocol

Through the SIP Event Framework [19], SIP user agents (UA) can subscribe to event sources and be notified about changes in the event state. This event framework is generic and requires that usages of it, so called event packages, be defined for specific applications of it, like presence as explained in the following paragraph.

The concept of presence [20] and how to handle it is defined by the IETF. Amongst other definitions, it defines a presentity (presence entity) as an entity that provides presence information to a presence service. An event package has been defined for presence [21], thereby allowing UAs to be notified about presence state changes. This role of the UA is known as a Presence Watcher. The most commonly used presentity

today is a person [22-23], although also solutions with other presentities, such as sensors, are possible. The 3GPP Presence Service [3] leverages this Presence Framework for use in IMS.

Service Presence

Currently, the IETF Presence data model [24] targets people's presence and communication devices. This can be seen from the definitions of "service" and "device" within that specification, where they are defined as people's communication devices with communication services. These definitions exclude many kinds of services, as a service does not necessarily need to do any communication in the context of a user. Moreover, in this model a service's presence is directly related to a user, and that is also not necessarily true for all services. For example, a temperature sensor in a user's house and a media server's presence does not need be included in a person's presence.

The vision with the concept of *Service Presence* is to extend the Presence data model with definitions for generic services. This allows a Presence Watcher to be aware of the presence of a service. Whereas presence information of a person includes elements such as mood, context and contact information, the presence information for a service mainly includes a description of the service.

Furthermore, this concept enables remote service discovery, transport and provision of service related information through SIP networks, such as IMS. Information about services discovered in one service network is transported via the common communication infrastructure to another service network (remote network). In the remote network the service presence information can be used to establish control sessions using [1]. The basic concept is illustrated in Fig. 1 and explained in the section below.

Architecture

Consider two local network environments, Residential network 1 and Residential network 2, as illustrated in Fig. 1. Each Residential network uses its own distinct communication and SDP mechanism, depending on the types of devices and services within the network. These residential networks are inter-connected, for example with an

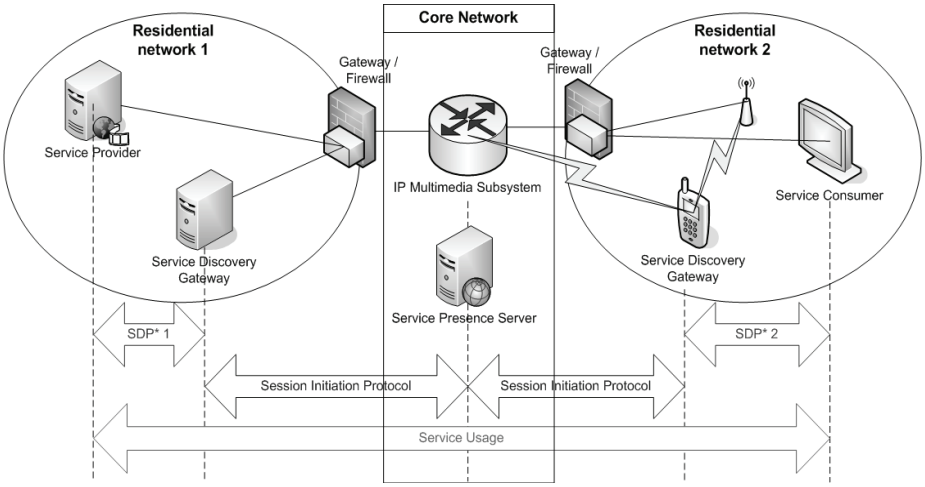


Fig. 1 Service Presence concept. Two service networks are connected through a common core network, IMS, and can share services. (*) SDP Service Discovery Protocol.

IMS enabled fixed and/or wireless access infrastructure. These two residential networks cannot access each other's services directly, as earlier explained. Therefore, an architecture for service presence is necessary. This architecture includes four logical nodes, as explained in the following.

Service

Presentity discovered through a service discovery mechanism. In Fig. 1 this can be seen as the "service provider" node.

Service controller

Service controller is an entity that discovers services, using a SDP, in the local network and can control them. This node is illustrated in Fig. 1 as "service consumer".

Service Discovery Gateway

The Service Discovery Gateway (SDG) is responsible for discovering services in its vicinity and making their presence available for remote access. Furthermore, it can subscribe to receive service presence information from other SDGs. This enables it to be aware of remote services as well.

To make presence information available it needs SIP UA functionality. This allows the SDGs to take advantage of SIP functionality, such as locating user agents. This allows services of residential networks to indirectly be part of SIP sessions as well.

Service Presence Server

To support the SDGs this node is introduced in the core network and collects presence state from them. This offloads the SDGs with the task of notifying all presence watchers. In addition, it can schedule command requests to avoid overloading the SDGs with requests to a service. This is especially important for residential services that often are not designed for high load usage.

As described later, in the *Deployment options* section, this gives two different deployment options for service presence.

SDP interoperability

As illustrated in Fig. 1 it is possible that different service discovery mechanisms are used within different residential networks. Accordingly, the service discovery gateways have to support different service discovery protocols SDP*1 and SDP*2. By translating the service discovery protocol specific service information into a generic service presence information format within each service discovery gateway, and exchanging this generic service presence information by means of the introduced service presence framework, it becomes possible that different service discovery mechanisms can cooperate in a way that the corresponding services can be utilized between different residential networks. Each service discovery gateway translates the received generic service information into the format that is used by the service discovery protocol within its network, and then publishes this information about the remote service within the local network in the format supported by the local devices.

From a control plane point of view, this interoperability functionality enables for example that a Bluetooth device (like a wireless headset) plays out media provided by a UPnP media server, given that the protocols of the underlying media plane are compatible.

Deployment options

In principle, two different deployments of the service presence concept are possible:

- *Peer to peer deployment:* In this case, the service discovery gateways within different residential networks exchange the SIP presence messages with the service presence information directly between each other. Therefore, no presence support from the operator core network is necessary.
- *Operator centric deployment:* With this option, the operator provides a service presence server that operates as service presence relay between service discovery gateways.

We consider the latter deployment option as the standard deployment, and it is used in the rest of this paper. The former option is good for few SDGs, but it scales fairly bad without further optimizations. Finally, the former option makes it possible to use service presence as an enabler for other services.

Remote Service Awareness Protocol

Our "PIRANHA" protocol presented in [1] has been extended with support for Service Presence. These extensions are based on the 3GPP Presence Service [3]. An example of the protocol signaling is shown in Fig. 2.

The set of extensions includes two actions: service presence publication and service presence subscription. In addition, it specifies an extension to the Presence Information Data Format [25] for describing services. These are described in the following sub-sections.

Service presence publication

"Session Initiation Protocol (SIP) Extension for Event State Publication" [26] differentiates between event hard and soft state. The main difference between them is that in the latter case the event state has a defined lifetime before it will expire, while in the former case the event state does not expire. Therefore, the former is used to describe

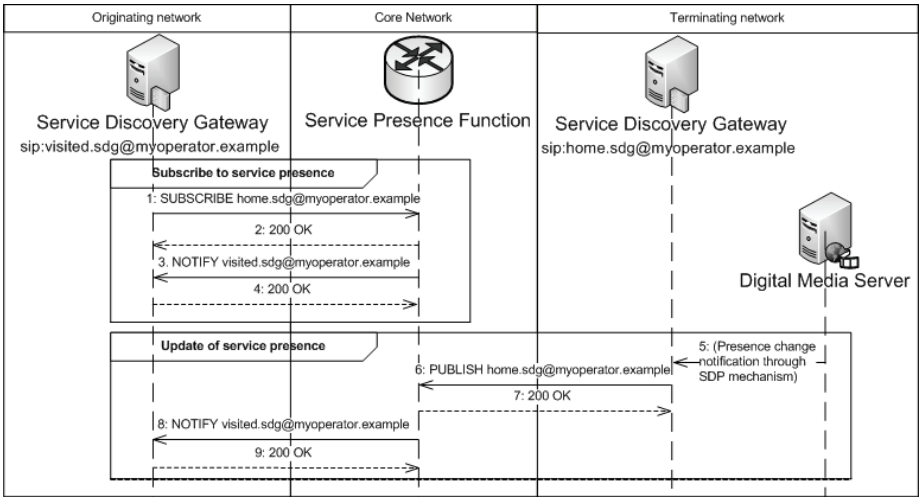


Fig. 2 Service presence signaling. An SDG in a visited network ("Visited SDG") subscribes to the services of another SDG ("Home SDG"). When the "Home SDG" updates the status of its services, the "Visited SDG" is notified by the Service Presence Function.

```
<?xml version="1.0" encoding="UTF 8"?>
<filter set xmlns="urn:ietf:params:xml:ns:simple filter">
 <ns bindings>
  <ns binding prefix="pidf" urn="urn:ietf:params:xml:ns:pidf"/>
  <ns binding prefix="utl" urn=http://schemas.agdermobilitylab.com/UTL/">
 </ns binding>
 <filter id="1234" uri="sip:home.sdg@myoperator.com">
  <what>
   <include type="xpath">/pidf:presence/pidf:tuple/upnp root
device/device/deviceType[urn:schemas upnp org:deviceType:MediaServer:1]</include>
  </what>
  <trigger>
   <changed from="CLOSED"
to="OPEN">/pidf:presence/pidf:tuple/pidf:status/pidf:basic</changed>
  </trigger>
 </filter>
</filter set>
```

Fig. 3 Subscription filter for being notified about media servers when they are switched on.

initial, or default, states, while the latter is used for transient states. Event hard states can be modified through [27], while event soft states can be modified using the SIP PUBLISH mechanism defined in [26].

In the service presence concept introduced in this paper, the event hard state is used for all basic services that are provided continuously by a service network. Changes in the status of these services and other service-related information about temporarily available services are made available through event soft states, such as a service status changing from offline to online. This way, resources are not wasted to refresh the state information of default services, while temporary changes of service states will expire when they are no longer in effect. After the temporary state expires, the service will fall back to the default state.

Furthermore, to reduce even more of the amount of information sent for service state changes, only the differences since the last publication are necessary to be sent. These resource savings are important since a service presence information document can become very large as it includes all information about a service and how to use it (actions it supports, service parameters, etc).

Moreover, another mechanism to increase the resource efficiency when acquiring these documents is content indirection [28]. With content-indirection, the client can download the service presence information document using a different access link than the one used for signaling. For example, mobile phones with both a cellular link and a Wi-Fi link can receive the signaling through the cellular link and download the service presence information document through the Wi-Fi link.

SDGs publish event states to the SPS. The SPS is responsible for notifying all subscribing SDGs about the new or changed service event states.

Service presence subscription

An SDG can subscribe to an SPS to be notified on the service presence information of services that are under control of another SDG. A subscribe request can include filters [29-30] that support queries in the W3C XQuery [31] language. These filters can be used to restrict notifications about service presence information with regards to particular service types and when the notifications should be sent. For example, notifications can be requested only from services of type 'media servers' when their status changes to 'online', as shown in Fig. 3.

In addition to these filters, the SPS can apply restrictions regarding the availability of services to an SDG, through a policy document specified in the Presence Authorization Rules [32] format. This document can be stored either in the SPS or in an OMA XML Document Management Server [33].

```
<?xml version="1.0" encoding="utf 16"?>
<presence
 xmlns="urn:ietf:params:xml:ns:pidf" entity="pres:sip:higa.home1@ims.ict fiesta.test">
 <tuple id="uuid:89665984 7466 0019 5b46 051c73783736">
  <status><basic>open</basic></status>
  <upnp root device xmlns="http://schemas.agdermobilitylab.com/ServicePresence">
   <device
    xmlns="http://schemas.agdermobilitylab.com/UTL"
    UDN="uuid:89665984 7466 0019 5b46 051c73783736"
    deviceType="urn:schemas upnp org:device:MediaServer:1"
    friendlyName="Media Server by TwonkyVision"
    manufacturer="" modelName="">
    <serviceList>
     <service
      controlURL="http://192.168.1.9:9000/ContentDirectory/Control"
      eventSubURL="http://192.168.1.9:9000/ContentDirectory/Event"
      serviceId="urn:upnp org:serviceId:ContentDirectory"
      serviceType="urn:schemas upnp org:service:ContentDirectory:1" />
     <service
      controlURL="http://192.168.1.9:9000/ConnectionManager/Control"
      eventSubURL="http://192.168.1.9:9000/ConnectionManager/Event"
      serviceId="urn:upnp org:serviceId:ConnectionManager"
      serviceType="urn:schemas upnp org:service:ConnectionManager:1" />
    </serviceList>
   </device>
  </upnp root device>
 </tuple>
</presence>
```

Fig. 4 Service presenence information document for a media server.

Service presence information document

In the body of the notification messages, sent in steps 3 and 8 in Fig. 2, the service presence information document is included. This document is formatted using the Presence Information Document Format (PIDF) [25] with extensions for describing services. In particular, each service is included as a presence information tuple containing the service status and a description of the service itself. An example of such a document is given in Fig. 4.

Notice that the document includes private IP-addresses. The consuming SDG needs to replace these when using the device. This can be done *after* establishing a service usage session.

Security and privacy

Extending the Presence Service to include information about devices and services implies that users share more information about their private environment with their operator. The metadata associated with a device is one concern, because it includes information about the manufacturer, product model and other details. For example, burglars might like to get this information to find out which locations are attractive for theft, especially if location information is included with the presence information. Moreover, residential security devices may be included as well. Knowledge of corresponding security services, and possibly even control them, would make burglars' job easier.

Furthermore, there are other possibilities for misuse of this information as well. For example, being aware of the media devices (as e.g. networked TV devices) people own, and getting information about how to access them from outside the home network, can make it easier to target them with advertisements.

Now, these security and privacy aspects are basically the same as those that apply to a person's presence data. Therefore, the same countermeasures used in the basic User

Presence service, such as Presence authorization rules [32], applies to service presence as well.

Prototype for Remote Service Awareness

For a proof of concept, a prototype of the service presence concept has been implemented. This prototype includes both the SDG and SPS nodes. The SDG functionality has been incorporated into our earlier prototype described in [1] to enhance it with service presence.

Both parts of the prototype were implemented utilizing the Ericsson Service Development Studio (SDS) 4.0.

SDG Application

The SDG application includes a Web-based management portal ("Service Management"), depicted in Fig. 5, that shows all available UPnP devices in the same local network as the SDG, and lets the user select ("Registered at SDG-Core" column in Fig. 5) which devices should be made available in SPS. Availability in the SPS is accomplished through service presence publication of the selected devices (and their services), as described above. The major components of the SDG application are described in the following sub-sections.

Presence Awareness Handling

This component keeps track of local devices / services, currently only UPnP. It allows other components to be notified when the service presence status changes. For example, both the Piranha and Service Management components receive such notifications.
HTTP Server: A lightweight HTTP server has been implemented to host the Service Management component and the UPnP devices by receiving SSDP (unicast) and SOAP requests.

Piranha

Implements the service presence protocol based on the Ericsson IMS Client Platform (ICP) API, which is part of the Ericsson SDS.

It uses the W3C Document Object Model (DOM) API for creating the service presence publication documents and parsing the service presence notification documents. The Apache Xerces [34] library has been used as an implementation of the W3C DOM API.

Service management website

As shown in Fig. 5, this web site allows users to see and manage which local services should be made available for remote SDGs, through the SPS. Also, it supports subscribing to service presence events notified by another SDG. These functionalities are realized through the Piranha and PAH components.

This functionality has been implemented using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript. On the server side, dynamic parts of the page, such as table content, are created by simply replacing well-known HTML-comments with the actual content.

"Service Presence Core" Service

The SPC Service has been implemented as a SIP Servlet [35], utilizing Ericsson SDS for testing and development. It implements the "Service Presence Server" functionality described in the *Service Presence* section above. In the following, the components of this application are described.

Service Presence Database

A simple SQL-database with tables for presentities, subscribers and presence publications.

Event Manager

Handles publications and subscriptions by storing them in the Service Presence Database, and schedules tasks to handle expiration of these.

To be able to notify subscribers, it also keeps session state (javax.servlet.sip.SipSession objects) for them in a separate table with an implicit relationship to the subscriber database table. The reason for this split of where state is stored is due to simplifying the implementation.

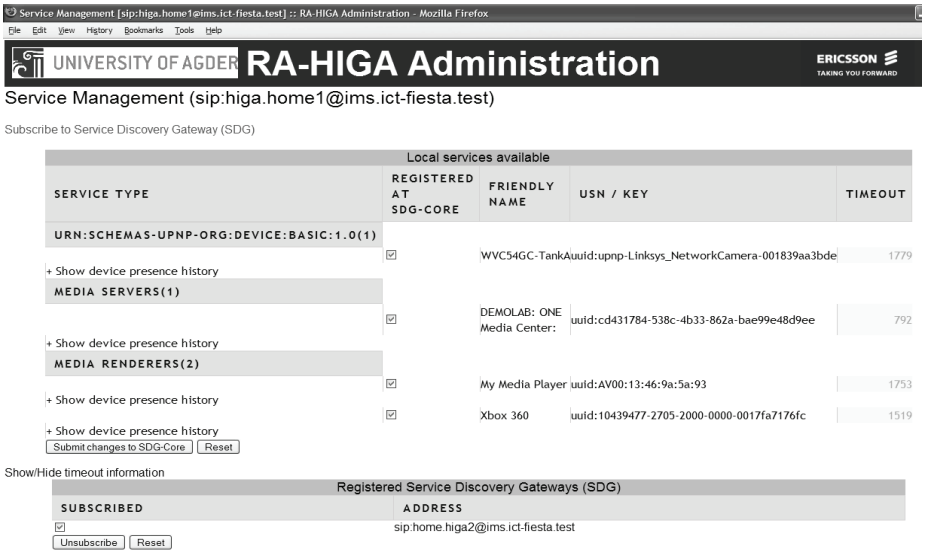


Fig. 5 Service management web page.

Service Presence Servlet

Handles requests and responses for PUBLISH and SUBSCRIBE requests using the Event manager component.

Deployment

The prototype has been deployed as illustrated in Fig. 1 with an IMS core network and two residential networks. The core network consists of the OpenIMS [36] implementation of the IMS core nodes and a Sailfin [37] application server that hosts the Service Presence Core service. In the residential networks, some UPnP devices (media server, media renderer and a gateway device) are available, in addition to a computer hosting the SDG.

Each SDG makes a NAT-binding in the local gateway device for its signaling with IMS, using UPnP, before registering with the core network. For security reasons, some gateways support that such NAT-bindings are bound to a remote host address that may use it to stop illegitimate access.

Results

With the prototype, we have found that the service presence protocol described in *Remote Service Awareness Protocol* above works as expected. The SDGs publish status information about the selected local services, and subscribers are notified when the status changes. For example, if a media renderer device is shutdown its presence status will be updated and subscribers are notified.

Conclusion

In this paper we have introduced the concept of service presence. With our prototype we have shown how it can be realized, and that the protocol works as expected.

This concept enables many new scenarios related to residential services, and services in general. For example, it allows Internet services to deliver content directly to residential services, such as media players.

References

[1] A. Häber, M. Gerdes, F. Reichert, R. Kumar, and A. Fasbender, "Remote Service Usage through SIP with Multimedia Access as an Use Case," *18th IEEE Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2007)*, Athens, Greece, 2007.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, 2002.

[3] 3GPP, "Presence service; Architecture and functional description; Stage 2", TS 23.141

[4] F. Zhu, M. W. Mutka, and L. M. Ni, "Service discovery in pervasive computing environments," *Pervasive Computing, IEEE,* vol. 4, pp. 81 90, 2005.

[5] S. Vinoski, "Service discovery 101," *IEEE Internet Computing,* vol. 7, pp. 69 71, 2003.

[6] A. Häber, "Service Discovery," in *Mobile Phone Programming: Application to Wireless Networking*, F. H. P. Fitzek and F. Reichert, Eds.: Springer Netherlands, 2007, pp. 239 255.

[7] S. Helal, "Standards for service discovery and delivery," *Pervasive Computing, IEEE,* vol. 1, pp. 95 100, 2002.

[8] G. G. Richard III, *Service and Device Discovery: Protocols and Programming*: McGraw Hill Professional, 2002.

[9] "UPnP Device Architecture," UPnP Forum, 2003. http://www.upnp.org/specs/arch/UPnP DeviceArchitecture v1.0.pdf

[10] M. Jeronimo and J. Weast, *UPnP design by example : a software developer's guide to universal plug and play*. Hillsboro, Or.: Intel Press, 2003.

[11] Apple, "Bonjour," 13 January 2008. http://developer.apple.com/opensource/internet/bonjour.html

[12] Bluetooth, "Specification of the Bluetooth System," 2nd ed, 2004. http://www.bluetooth.com/NR/rdonlyres/1F6469BA 6AE7 42B6 B5A1 65148B9DB238/840/Core_v210_EDR.zip
[13] "UPnP AV Architecture," 12 December 2007. http://www.upnp.org/specs/av/UPnP av AVArchitecture v1 20020622.pdf
[14] A. Häber, F. Reichert, and A. Fasbender, "UPnP Control Point for Mobile Phones in Residential Networks," in *15th IST Mobile & Wireless Communication Summit* Myconos, Greece, 2006.
[15] "Home Networked Device Interoperability Guidelines, v1.0: An Industry Guide for Building Interoperable Platforms, Devices and Applications," Digital Living Network Alliance, 2004.
[16] "DLNA : Home : Enjoy your music, photos and videos, anywhere anytime," 12 December 2007. http://www.dlna.org/
[17] G. Camarillo and M. A. García Martín, *The 3G IP multimedia subsystem (IMS) : merging the Internet and the cellular worlds*, 2nd ed. Chichester: Wiley, 2006.
[18] C. Gourraud, "Using IMS as a Service Framework," *Vehicular Technology Magazine, IEEE,* vol. 2, pp. 4 11, March 2007 2007.
[19] A. B. Roach, "Session Initiation Protocol (SIP) Specific Event Notification," RFC 3265, 2002.
[20] M. Day, J. Rosenberg, and H. Sugano, "A Model for Presence and Instant Messaging," RFC 2778, 2000.
[21] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)," RFC 3856, 2004.
[22] H. Schulzrinne, V. Gurbani, P. Kyzivat, and J. Rosenberg, "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)," RFC 4480, 2006.
[23] H. Christein and P. Schulthess, "A General Purpose Model for Presence Awareness," in *Distributed Communities on the Web: 4th International Workshop, DCW 2002, Sydney, Australia, April 3-5, 2002. Revised Papers*, 2002, pp. 345 392.
[24] J. Rosenberg, "A Data Model for Presence," RFC 4479, 2006.
[25] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, and J. Peterson, "Presence Information Data Format (PIDF)," RFC 3863, 2004.
[26] A. Niemi, "Session Initiation Protocol (SIP) Extension for Event State Publication," RFC 3903, 2004.
[27] "XML Document Management (XDM) Specification," Open Mobile Alliance.
[28] E. Burger, "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages," RFC 4483, 2006.
[29] H. Khartabil, E. Leppanen, M. Lonnfors, and J. Costa Requena, "Functional Description of Event Notification Filtering," RFC 4660, 2006.
[30] H. Khartabil, E. Leppanen, M. Lonnfors, and J. Costa Requena, "An Extensible Markup Language (XML) Based Format for Event Notification Filtering," 2006.
[31] D. Chamberlin, "XQuery: a query language for XML," *Proceedings of the 2003 ACM SIGMOD international conference on Management of data,* pp. 682 682, 2003.
[32] J. Rosenberg, "Presence Authorization Rules," RFC 5025, Dec 2007
[33] "XML Document Management Architecture," Open Mobile Alliance 2006.
[34] Apache Software Foundation, "Xerces2 Java parser," Apache XML Project. http://xerces.apache.org/xerces2 j/
[35] A. Kristensen, "SIP Servlet API Specification," *Java Specification Request (JSR), Java Community Process,* vol. JSR 116, 2002.
[36] T. Magedanz, D. Witaszek, and K. Knuettel, "The IMS playground @ FOKUS an open testbed for generation network multimedia services," 2005, pp. 2 11.
[37] java.net, "Sailfin: SIP Servlet Container," 15 January 2008. https://sailfin.dev.java.net/

Virtualization of Remote Devices and Services in Residential Networks

Andreas Häber, Jesús Gómez Ruiz De Mier, and Frank Reichert
University of Agder
Faculty of Engineering and Science
Grimstad, Norway
{andreas.haber | frank.reichert}@uia.no

***Abstract*— Lately solutions for remote access for residential services have been proposed. However, these solutions require modifications to the service controllers. In addition, remote access adds complexity to the client application. We propose here a solution for decoupling remote access from the client itself with an entity that creates virtual instances of remote services in a local network. Thereby, clients will be able to discover the virtual instance and use it. Moreover, client applications do not need to distinguish between local and remote services hence reducing complexity.**

Keywords- connected home, remote access, virtualization, UPnP

I. INTRODUCTION

RESIDENTIAL services, including entertainment and communication appliances, are increasingly being digitalized and interconnected by standards such as Universal Plug & Play (UPnP). For example, digitalized photo albums can be viewed from any media playing service in a network, such as TVs and digital photo frames.

Lately, several solutions have been proposed to allow these services to be used remotely as well. Remote access let users access their services when visiting friends, on the road and elsewhere. As shown in [1], users can use remote access to show their photos stored at their home media server when visiting friends, without for example going by any third party photo sharing site like Flickr or Picasa.

However, the current remote access solution proposals require enhanced control points that can be aware of both local and remote devices at the same time. There are two shortcomings with these solutions. Firstly, standard control points are not supported, and thus cannot be used to control remote devices. Because there are already several products out in the market, including TVs, PC applications and media players, it would benefit consumers who have already invested in this equipment if they can seamlessly work with remote devices. sThe second drawback is that they make control points more complex by requiring them to keep track of both remote and local devices using different protocols. For instance, with [2] the control point needs to use web syndication protocols to be aware of remote resources and the Simple Service Discovery Protocol (SSDP) to be aware of local resources. This separation increases the complexity of control point software.

In this article we show how both of these drawbacks can be amended by virtualizing remote services in a network. Virtualization of remote services means that they will appear in a network as if they physically exist there to clients (e.g. control points in UPnP terms). Although we use UPnP as an example in this article we believe the same concept can be applied to other service discovery protocols as well, such as Bonjour and Bluetooth. Fig. gives an example of virtualization, where users can enjoy content from a remote media server using the digital media player of

Our concept requires a function, named *service virtualizer*, that has remote access capabilities and can present the remote services using a service discovery mechanism for local control applications. This function can be deployed at for example residential gateways, as shown in Fig. 1, mobile terminals and car infotainment systems. In general, control requests and event notifications have to be proxied to and from the remote service by the virtualizer as well. However, if the service description for a service includes metadata on its operations, then the virtualizer can cache results and return them for future invocations of that operation. For example, services often have properties that are constant until it is restarted, such as supported capabilities.

This article is organized in the following way. First, use cases for service virtualization are presented. Next, it is described how service virtualization reduces complexity of control points. Thereafter, we present our solution for service virtualization. Then a prototype implementation of our solution design is described. Finally, we give conclusions on our findings.

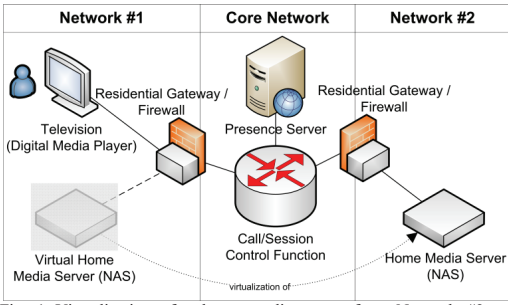


Fig. 1 Virtualization of a home media server from Network #2 to Network #1, by the residential gateway. The virtual home media server can be used by a standard digital media player in Network #1.

II. SCENARIOS

In this section we present three use cases enabled by service virtualization. At the end of the section a short analysis of advantages and disadvantages is given.

A. Remote multimedia access

Fernando is visiting his girlfriend Amanda, and she wants him to show her pictures of a trip he did lately. Her TV has inbuilt media player functionality, but unfortunately Fernando did not bring the photos with him. However, using his mobile phone he can virtualize his home media server in her network, and thereby access it from the TVs control interface.

B. Remote multimedia storage

Just before leaving Amanda's place, Fernando remembers that a new TV show that he wanted to watch starts. Since he lives an hour away from her it will be over before he reaches home. Therefore, he asks her to record the TV show to his virtualized media server, working like a personal video recorder. This way, the TV show will be waiting for Fernando when he reaches home.

C. Remote control

On the train home from Amanda, Fernando wants to adjust the air conditioning system of his apartment before he arrives home. He starts the air condition control application on his mobile phone. By virtualizing his air conditioning system on the loopback adapter he ensures that other people on the train won't be able to discover and take over his air conditioning system.

D. Scenario analysis

The first two scenarios show that the decoupling service virtualization gives allows clients to consume remote services without modifications to themselves. Amanda's TV did not need any extra software for using the remote media server. Rather, it is handled transparently by the virtualizator. Especially devices, such as digital media players, that only support the 2 box model benefit from this. These devices would otherwise not be able to discover remote media servers without upgrades.

In the third scenario it is shown how service virtualization simplifies applications. Because the air condition control application did not need to handle remote access to the air condition system, its application designers can rather focus on creating a good user interface for controlling air condition systems.

III. SERVICE VIRTUALIZER ARCHITECTURE

Our proposed architecture of the service virtualizer, as shown in Fig. 2 has been designed to be decoupled from users of the virtualized services and give a low overhead. In the following we describe the components and then how they operate together.

A. Architectural components

The service virtualizer is based on the *service discovery gateway* function introduced in [3]. This function is aware of remote services by subscribing to a remote peer. Either it subscribes directly to another service discovery gateway or by an intermediate server, such as a presence server. It exposes an interface, Piranha status, which allows internal components to be notified of remote service changes. Naturally, this interface is used by the service virtualizer to be aware of remote services.

Based on the available remote services, the *remote service virtualizer* creates virtualized instances of them. It is responsible for service discovery and description of these services using the supported service discovery protocols. For instance, for UPnP it must multicast NOTIFY messages, handle M SEARCH requests from control points and reply to requests for retrieving device and service description documents. Therefore, it must be able to create these documents based on information from the service discovery gateway. Notice that it cannot simply use the same description documents from the real device, because they include details, such as where to send control requests, that won't work with the virtual device. Instead, the description documents refer to its control request and event proxies.

Two proxies, the *control request proxy* and the *event proxy*, handle control requests/responses and notifications to and from the real device. These proxies optimize performance and balance load for the remote service. For instance, only one event subscription to the remote service is required instead of each client has its own subscription. Furthermore, because control and event messages may include private IP addresses that must be changed to public IP addresses. Otherwise, the message receiver won't be able to reach the remote host. Perhaps the worst case is when the local and remote networks use the same address space, for example 10.0.0.0/24. In that case control points using the virtual device would try to connect to a local host instead of the remote host. For example a control point may request a local media player

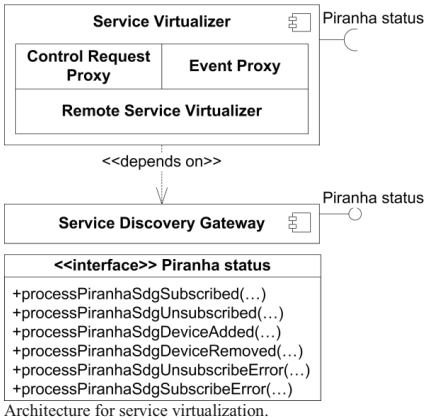


Fig. 2 Architecture for service virtualization.

to play content from a virtual media server. If the content refers to a local host then the media player will try to connect to that host to retrieve the content. Instead of changing the message content as proposed here, “Home DNS” [4] by P. Belimpasakis et al. offers a more advanced solution using DNS that should give better performance.

B. Operation of the service virtualizer

In the following we describe the operation of the service virtualizer for UPnP devices. Five operation sequences are covered, as shown in Fig. .

The first sequence shown, remote service discovery, is a pre requisite based on the presence service extensions for remote service discovery described in [3]. The local service discovery gateway subscribes to a presence server and receives presence notifications when the remote service discovery gateway publishes updates.

When a service discovery gateway is notified of presence state updates, the service virtualizer is notified through the processPiranhaSdgDeviceAdded method at the internal Piranha status interface. This is shown in the *device* virtualization sequence in Fig. . Several approaches are possible for the service virtualizer to know which devices it shall virtualize. The simplest approach is on demand by a user. Another option is an administrator managed list that will be checked against to see whether a remote device should be virtualized. Virtualization of a device is carried out similarly to normal hosting of a UPnP device. Information obtained from the PIDF document is used when creating the virtual device. Care must be taken so that all references, such as where to send control requests, subscribing to events, and obtaining device and service description documents, all point to the service virtualizer instead of the remote device. The immediate result of the virtualization is that the virtual device is announced in the local network with a SSDP NOTIFY message. Moreover, SSDP search requests will be replied to for virtual devices.

After a control point is aware of a new device it may subscribe to receive events and send control requests in any order. Although these are different operations both of them requires interaction with the remote device. Therefore, a remote service usage session [5] must be established through the service discovery gateway. By establishing a usage session the remote service discovery gateway configures its local firewall and routing tables to allow requests from the requestor. As discussed above, private IP addresses in messages to and from the remote device must be must be carefully handled by the service virtualizer.

Event subscription requests by local service controllers for, virtual devices are sent to the service virtualizer. The requests are sent there because the device description documents refer to the service virtualizer instead of the remote device. Therefore the service virtualizer can aggregate event subscriptions to reduce external traffic.

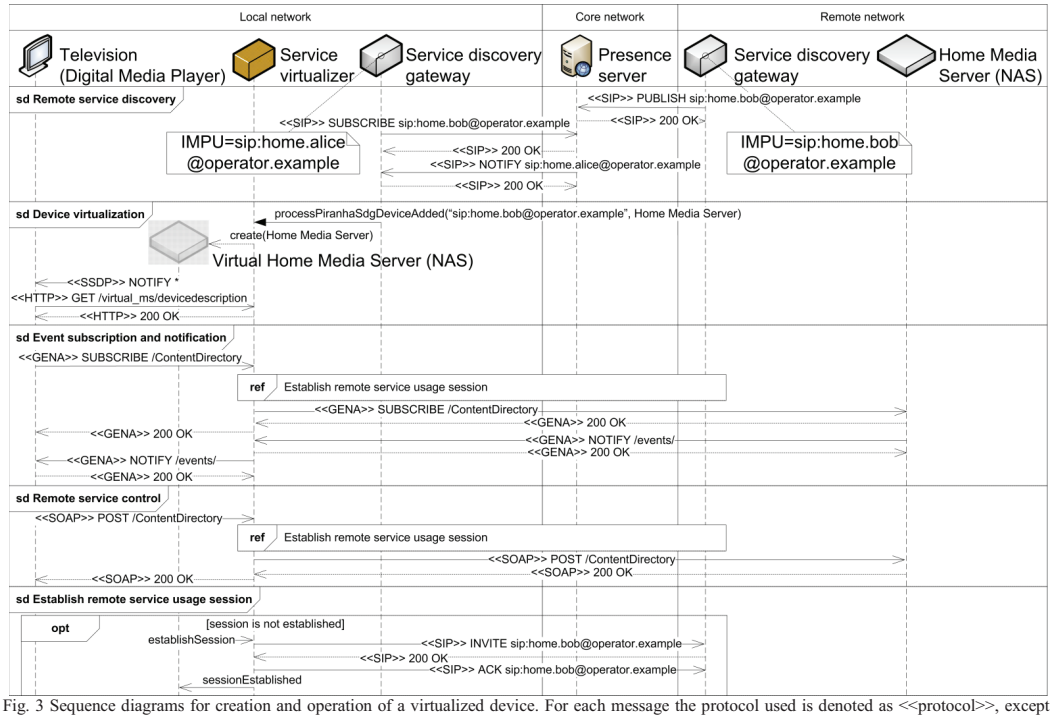


Fig. 3 Sequence diagrams for creation and operation of a virtualized device. For each message the protocol used is denoted as <<protocol>>, except internal messages passed between the service virtualizer and the service discovery gateway in the local network.

For the first event subscription request the service virtualizer will subscribe to the remote device. Since traffic to remote networks cost more then local traffic, the service virtualizer should request a long subscription duration (e.g., 3 hours). Successive event subscription requests will be handled locally by the service virtualizer. The UPnP event notification protocol, Generic Event Notification Architecture (GENA), requires that devices send an initial event message to new subscribers. Because this message shall include the current state for all evented variables the service virtualizer has to keep track of the current state from the event notifications received. When the service virtualizer receives event notifications from a remote device it will send these to the local subscribers.

Because control requests often alter the state of the remote device, the service virtualizer cannot handle them directly. Therefore it has to proxy those requests to the remote device.

Not included in Fig. 3 is removal of virtualized devices, so called devirtualization. For example, if the presence of the remote device becomes offline the service virtualizer should remove the virtual instance. Another case is if the administrator of the service virtualizer requests a virtual device to be removed. To devirtualize a device, the service virtualizer should first tear down the virtual representation by stopping to both announce the device and to respond to search requests for the device. Moreover, the remote service usage session should be teard down by sending a BYE request to the remote service discovery gateway.

IV. SERVICE VIRTUALIZER PROTOTYPE

We have implemented a service virtualizer prototype that covers the functionality described above except the eventing proxy. It has been implemented as part of a Java 2 Micro Edition (J2ME) library. This library can be used by applications deployed to for example mobile phones and residential gateways. It uses the Ericsson IMS Client Platform to handle IMS communication.

A. Test configuration

The service virtualizer has been tested together with two residential networks connected together by a common backbone, similar to Fig. . However, the service virtualizer and the service discovery gateway functions are co located in a separate node from the residential gateway in both network #1 and #2. In network #1 a D Link DSM 320 Wireless Media Player is used as a digital media player, and in network #2 a Synology DiskStation 108j is used as a media server. OpenIMS [6] is used for the core network, in addition to a Sailfin [7] application server that hosts the presence server.

B. Results

After being notified about a remote device that shall be virtualized the service virtualizer creates a virtual device and announces it in the local network. We have verified that this works by receiving requests for the description documents of the virtual media server from the digital media player. The digital media player can only know the location for the device description documents by receiving proper SSDP messages from the service virtualizer. Therefore, requests for the description documents show that the digital media player has transitioned from step 1, device discovery, of UPnP to step 2, device description.

Moreover, control requests are handled by the service virtualizer by swapping IP addresses in the messages. However, this operation takes a considerable amount of time, and we therefore strongly recommend using Home DNS for handling private IP addresses instead.

V. RELATED WORK

An important enabler for service virtualization is remote service discovery. One approach is given in [2], where the Atom publishing protocol is used to allow peers to exchange presence information for local services. [3] achieves the same goal by extending the SIP Presence framework for services. The main benefit of [3] is that the Presence service has already been introduced to IMS.

VI. CONCLUSION

In this paper we have proposed virtualization of remote devices as a solution to enable non remote access capable service controllers to also interact with such remote devices. Through rich service presence information the proposed service virtualizer function takes care of local discovery and description requests for remote devices.

Moreover, this virtualization technique can also simplify remote access capable clients. Applications won't have to distinguish betwenn local and remote devices, because the virtualization makes remote devices appear as local devices. Therefore, both the service discovery gateway and the service virtualizer functions should be part of a client's middleware platform.

The eventing proxy component of the service virtualizer aggregates eventing for all local service controllers. In particular, this reduces the communication costs for the remote device to send notifications. Because several actions do not change the state of a service it is desirable to allow the service virtualizer to reply to such requests, instead of proxying it to the remote devive. However, this requires that the service virtualizer is capable of knowing which actions can be considered safe, in addition to the current state of the related state variables. The former of these require additional metadata in the service description documents for UPnP. With regards to the latter requirement, assuming that such state variables are evented, the current state will be obtained by keeping track of the event notifications from the service.

REFERENCES

[1] P. Belimpasakis, S. Moloney, V. Stirbu, and J. Costa-Requena, "Home media atomizer: remote sharing of home content - without semi-trusted proxies," *Consumer Electronics, IEEE Transactions on,* vol. 54, pp. 1114-1122, 2008.

[2] P. Belimpasakis and V. Stirbu, "Remote Access to Universal Plug and Play (UPnP) Devices Utilizing the Atom Publishing Protocol," *Networking and Services, 2007. ICNS. Third International Conference on*, 2007, pp. 59-59.

[3] A. Häber, M. Gerdes, F. Reichert, A. Fasbender, and R. Kumar, "Using SIP Presence for Remote Service Awareness," *Norsk Informatikkonferanse 2008 (NIK'08)*, Kristiansand, Norway, 2008.

[4] P. Belimpasakis, A. Saaranen, and R. Walsh, "Home DNS: Experiences with Seamless Remote Access to Home Services," *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, 2007, pp. 1-8.

[5] A. Häber, M. Gerdes, F. Reichert, R. Kumar, and A. Fasbender, "Remote Service Usage through SIP with Multimedia Access as an Use Case," *18th IEEE Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2007)*, Athens, Greece, 2007.

[6] T. Magedanz, D. Witaszek, and K. Knuettel, "The IMS playground @ FOKUS-an open testbed for generation network multimedia services," 2005, pp. 2-11.

[7] java.net, "Sailfin: SIP Servlet Container," 15 January 2008. https://sailfin.dev.java.net/

Phone-controlled Delivery of NGN Services into Residential Environments

Andreas Fasbender[1], Stefan Hoferer[2], Martin Gerdes[1], Takeshi Matsumura[3],
Andreas Häber[4], Frank Reichert[4]
[1]*Ericsson Research, Ericsson GmbH, Germany,*
[2]*Department of Communication and Distributed Systems, RWTH Aachen, Germany,*
[3]*Ericsson Research, Ericsson Nippon K.K., Tokyo, Japan,*
[4]*Agder Mobility Lab, University of Agder, Grimstad, Norway*

Abstract

The horizontally layered architecture of the IMS/NGN standards family enables the delivery of services independent of access network and requesting device. In this article, the authors propose a further separation of service control and delivery, allowing the requesting device – in particular a user's mobile phone – to invite other devices (we will focus on DLNA appliances) into the service delivery, enhancing both user experience and service design flexibility. The proposed solution builds on exploiting proximity technologies (e.g. barcodes, NFC) for pairing the control device with a remote environment. Motivated by scenarios, the architecture concepts are explained and a prototype that was implemented for validation is described. Selected findings and a short overview of related standardization efforts conclude the paper.

1. Introduction

Entertainment devices such as set-top boxes, game consoles, music players, and cameras today routinely come with built-in networking capabilities that enable them to upload, download, and render media from other devices in the home. The Digital Living Network Alliance (DLNA) is since 2004 publishing interworking guidelines for home media sharing services [1] based on the Universal Plug and Play (UPnP) standards family [2]. DLNA is now widely accepted in the consumer electronics industry and will soon enable advanced interworking services for all sorts of devices in (local) IP network islands.

In parallel, fueled by a rapidly growing broadband penetration both in fixed and mobile scenarios, consumers are increasingly adopting online media download and streaming services such as music portals, mobile TV and fixed IPTV. Operators on the other hand have started to prepare for an increasing media mix by rolling out next-generation network (NGN) infrastructures and services based on IP Multimedia Subsystem (IMS) for service control and IP transport. Standardization bodies, such as the 3rd Generation Partnership Project (3GPP) [3], Open Mobile Alliance (OMA) [4] and the Open IPTV Forum [5], are specifying basic services and enablers to deliver operator-managed and 3rd party services via this infrastructure.

Many telecommunication services today have in common that they are designed and optimized for a single consumption device, for example a mobile phone, an IMS Multimedia Telephony (IMT) terminal, or a set-top box (STB). Devices are typically securely coupled to the user's identity and subscription by Subscriber Identity Module (SIM) cards or conditional access modules, consequently restricting service delivery to a specific consumption device and often even location. Mechanisms such as placeshift (e.g. Orb, Slingbox) have to be put in place to support the user's growing demand for access to content and services, everywhere and anywhere.

The user's phone as a personal device holding the identity, service portfolio and personal data (such as address book, media files, and service credentials) is today heavily under-utilized as a service control device, one important reason being limitations in screen size and input facilities. In this paper, we propose a new service delivery concept, which relaxes the tight coupling of service control and delivery through the use of IMS, and allows users to initiate and control their services on for example a mobile phone, while delivering the services to a most suitable consumption device. This leads to a triangular relationship between user, the user's services (aggregated from multiple sources), and the devices used for service play-out and

978-0-7695-3333-9 /08 $25.00 © 2008 IEEE
DOI 10.1109/NGMAST.2008.71

interaction. Our proposed architecture combines the benefits of operator-guaranteed trust, security, charging and quality of service based on NGN technologies with the consumer electronics (CE) industry perspective of launching attractive end user devices.

We start by describing sample scenarios in Section 2, motivating the requirements on a flexible end-to-end solution. In Section 3 we present our proposal for an architecture that addresses these requirements, providing a description of all functional elements required and explaining the signaling flows. Our prototype media portal implementation, based on IMS, DLNA and QR Codes for proximity detection, is described in Section 4, followed by a summary of lessons learned and a short overview of ongoing standardization activities in related areas in Section 5. Section 6 concludes the article and points to open issues and potential future research.

2. Scenarios & requirements

2.1 Example scenarios

Remote music access: Carol is on a business trip, visiting a conference. On the way to the hotel she accesses the media portal of her service provider to listen to music with her mobile phone. After she has checked in, she decides to listen to her music in her hotel room and connects to her media portal again. Using her mobile phone she discovers all available media devices in the hotel room, with the stereo system and a TV set among them. This time she wants to enjoy the better sound quality of the hotel stereo system. Therefore, she selects the stereo system as target device for the music from the media portal, and her songs are immediately played on the stereo system in her room.

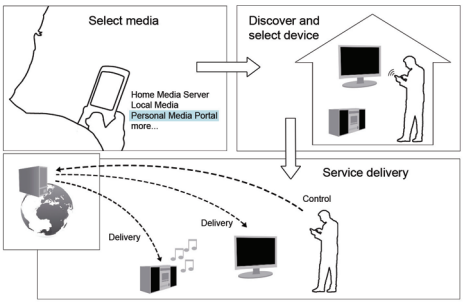


Figure 1. Separating service control and delivery

Remote DVR and Placeshift TV: The next day, Carol realizes that she will probably miss her favorite TV program in the evening. The *Placeshift TV* feature of her home IPTV subscription would actually allow her to redirect the TV media delivery from her IPTV provider through the hotel network and her room TV. But she expects to return late after the conference dinner. Therefore she prefers to log into her own residential control device at home from her phone, and to program her digital video recorder (DVR) with a few simple clicks to record the TV program for her.

2.2 Requirements

Considering above examples and similar user scenarios, a number of requirements for a widely applicable solution have been identified [6]:

For delivery of user services to devices in a remote environment a trust relationship between the user's identity, the service provider and the remote device selected for service consumption needs to be established. This relationship shall not depend on the source of the service, such as the user's home network, an operator application server, a 3rd party service provider, or the mobile device itself.

The intended solution should support interaction with any kind of remote device, such as UPnP/DLNA or SIP devices. Modifications to the software and hardware environment and the behavior of these devices shall not be required.

In order to provide an acceptable user experience, the user shall not be required to have any deep networking knowledge. Consequently, the user shall not be required to enter lengthy addresses, user names or passwords on the mobile phone or any remote device, an inconvenient and time consuming task.

For both, the network owner and its users, security is an important aspect of an acceptable solution. Most consumer appliances, including DLNA devices, lack a proper security implementation due to their restricted use in local network environments. Disclosing device information and other details shall only be allowed to trusted external peers in our solution.

The administrator of the visited network shall be able to grant access to selected devices and services, and restrict access for visitors. It must be possible to revoke access to any device at any point in time.

3. Architecture

The proposed architecture for the phone-based delivery of NGN services into residential environments is based on the following main principles: Connectivity and accessibility information about residential devices and their services is published to a presence server. A URL is transmitted to the mobile phone, pointing to the

presence instance where the connectivity and accessibility information for the residential devices can be retrieved. This URL is forwarded to application servers or other peers that subsequently use it for requesting detailed device and service descriptions. These details are then utilized for establishing a service delivery session into the residential network, using the phone for service control.

In Figure 2, the logical components of our architecture are illustrated. Functionalities and signaling flows are explained subsequently, under consideration of the *remote media access* use case. The signaling flows may vary in certain details for other use cases, but the same general principles are applicable.

3.1 Functional architecture

Because UPnP and similar service discovery protocols are designed to work in local IP networks, a *Residential Control Device* is necessary to make external nodes aware of the status and capabilities of devices within the local environment. In addition, the Residential Control Device must manage the access and connectivity of these devices through the residential gateway. Thus, the Residential Control Device allows using devices from the local network with external services, such as media delivery services from a portal to a local media player.

Essentially, the Residential Control Device provides the following functionalities (compare Figure 2): A *DLNA Control Point* (DLNA CP) is used to discover DLNA devices such as Digital Media Renderers (DMR) or Internet Gateway Devices (IGD) within the residential network. Corresponding device profiles are exposed from DLNA devices like TVs and music players, while the IGD device profile is provided by the residential gateway. After a device has been discovered, more details about device capabilities and supported services can be fetched from the respective device. This information is later required to access and control the offered services. To present the information on a DMR (e.g. in form of a barcode image shown on a DLNA TV) a *HTTP server* may also be deployed that serves as source for this information.

An *IMS Client or B2BUA* registers the Residential Control Device in the IMS core and hence connects the residential network to the NGN service infrastructure. This is used to publish detailed device and service information from the residential environment to the NGN Presence Server and to establish a secure and QoS enabled media tunnel between the *Application Server* and the residential control device for the service delivery into the residential environment. The B2BUA in the Residential Control Device also supports the handling of inbound session requests to SIP devices (or nodes with SIP UAs) within the residential environment [7].

The *Residential Environment Control Logic* contains the use case dependent functionality for the publication of device and service connectivity and accessibility information, and for the control of inbound service delivery sessions. It includes a management console for the selection of DLNA devices that are made available to the user for the delivery of an NGN service from the external service network, creates NAT-bindings at the residential gateway for inbound service delivery, publishes the required information for the inbound service delivery to a presence server, and transmits a reference to this information to the mobile phone. Different options can be supported for this transmission, including NFC, Bluetooth, or 2D barcodes displayed on the DLNA-TV and decoded by the mobile phone.

The *User's Phone* is the central service access and control device. It hosts an *IMS Client* required for

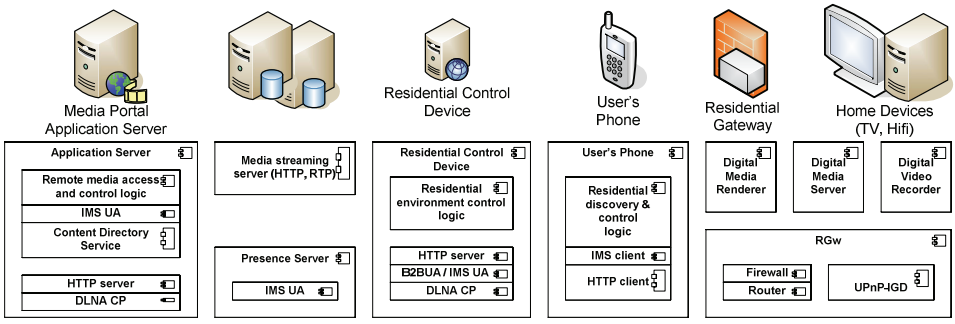


Figure 2. Functional end-to-end architecture

authentication to the NGN, accessing the NGN Application Server (the media portal in our example use case), and forwarding the reference to the device and service information (that has been published to a presence server) to the NGN Application Server. The *HTTP client* is used for any HTTP-based service control GUIs provided by an NGN application server. Through this control GUI the actual service delivery (e.g. streaming to a DMR) is decoupled from the service control on the user's phone. The *Residential Discovery & Control Logic* retrieves the device and service information reference from the residential control device and forwards it to the Application Server.

In order to deliver services into a residential network, the *Application Server* offers the following functions: It handles user authentication and authorization for personalization, coupling between user's service control point such as mobile phone and the service delivery target device, delivering the requested service to the target device securely with appropriate quality of service, and optional charging for the service. In our architecture, it hosts the service portal as the entry point for a user to select and request a service from the user's personalized menu. Through the IMS UA, it also implements a SIP interface to the IMS core over a standard ISC interface (IMS Service Control). Before allowing the user to access the services, the AS authenticates the user and authorizes service requests. Here, the IMS based architecture takes advantage of the Generic Bootstrapping Architecture (GBA) mechanism [8] to provide a single sign on (SSO) experience to the user. The DLNA Control Point (CP) controls UPnP/DLNA devices by sending UPnP actions to them over a secure tunnel.

A *Remote Media Access and Control Logic* establishes a secure tunnel used by the DLNA CP (IMS remote access, [9]). The residential network can delegate authentication of the AS or the user requesting access to it to the IMS network and authorize remote access for service delivery based on the authentication result. The *Content Directory Service* provides content lists such as video files or music albums that the user can watch or listen to. It also provides search functionality so that the user can easily find the desired content.

In the following we briefly explain the application of GBA mechanism to this architecture [8]: The IMS operator deploys a Bootstrapping Server Function (BSF), and the AS works as Network Application Function (NAF). If a User Equipment (UE) requests a service from the AS for the first time, the AS will demand that the UE must be authenticated using GBA. Thereafter, the UE and the BSF mutually authenticate using a shared secret. As a result, a pair of session keys is generated by the BSF, and one of the keys is delivered to the UE. The UE responds back to the AS with the received session key, where after the AS requests the BSF to authenticate the user by providing the session key. The BSF returns the authentication result and finally the AS approves that the UE is authenticated. Besides high security, this process has the advantage that it can be completed without the user having to type in a password.

Another component of the operator NGN infrastructure is a *Presence Server* that operates as information relay for device and service connectivity and accessibility information between the residential environment and the application server, which requires support for enhanced presence information formats.

Finally, a *Media Streaming Server* provides media storage and delivery through HTTP or RTP streaming.

3.2 Service delivery issues

In standard home environments, devices are connected to a residential gateway, which provides a mapping between the private IP address space used in the home and the publicly routable IP address space. In order to deliver end user services to residential devices, those devices first have to be known by the service provider and also be capable of interacting with the service. One option for device discovery is to rely on direct connectivity to the residential environment (e.g. using WiFi or Bluetooth). Besides the fact that not all mobile devices may be capable of or allowed to directly connect to the LAN, another disadvantage is that continuous listening for presence updates would drain the batteries of a mobile device rather quickly.

In any case, mobile devices that only have access to cellular connectivity need to discover local device information by other means. For this purpose, new proximity technologies such as barcodes or Near Field Communication (NFC) can be used. Here, the information is retrieved by reading so-called tags attached to the remote environment. To prevent the user from identifying each available device by a separate tag, it is reasonable to expose a central device and service repository located in the remote environment.

Furthermore, the lack of direct connectivity to the local network requires a solution for the user to control local devices via the service backend. This requires

mechanisms to traverse firewall and NAT in the residential gateway. One possible solution is to use the port management mechanisms offered by the UPnP Internet Gateway Device (IGD) profile [10]. IGD is widely deployed on off-the-shelf gateways; however, due to inherent security flaws it is not always available.

3.3 Signaling flows

High-level signaling flows are provided in Figure 3 to the right, consisting of three main phases as described in the following. Parts of the standard signaling with the IMS core has been omitted from the figures and explanations in the following, including the procedures for authentication and registration of the IMS entities.

In the *service presence publication* phase, the Residential Control Device publishes the presence information of devices and services in its local environment to the Presence Server.

In the *media selection and service awareness* phase, when the phone retrieves the initial page of a service, it passes an argument for the IMPU of the service presentity. This IMPU is acquired by the phone using a proximity solution, such as NFC or barcodes. Next, the Remote Media Access and Control Logic node receives the service information through having subscribed to the presentity of this IMPU. In case of static setups this flow can be optimized, e.g. by using a so-called one-shot SUBSCRIBE (expiration set to 0). Thereafter, media and playout device selection is carried out between the user's phone and the application server.

In the *remote service usage* phase, the Remote Media Access and Control Logic node establishes a remote session with the Residential Control Device, including the opening of a port at the Residential Gateway firewall for the media playout. Using this port mapping, the user can control the Digital Media Player with his phone through a GUI provided by the Remote Media Access and Control Logic. In the final step, the Digital Media Player fetches the content from the Media Streaming Server.

4. Prototype implementation

A simplified prototype based on the described architecture has been developed in collaboration of Ericsson Research, University of Agder and RWTH Aachen. This prototype supports a scenario where the user is a guest at a hotel providing Digital Media Players and retrieves media from a Media Portal. The prototype focuses on the implementation of the service delivery functionality and builds on HTTP signaling for remote service awareness instead of SIP/IMS.

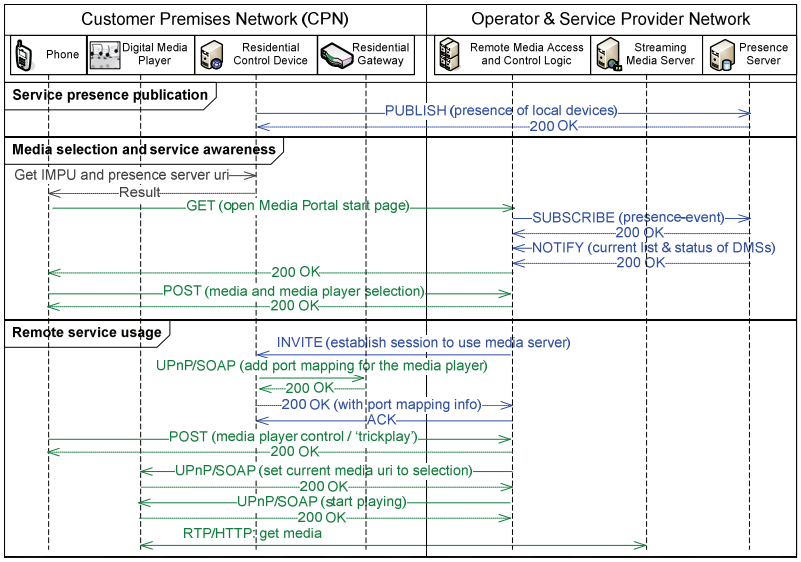


Figure 3. High-level signaling flow for remote media access on DLNA renderers

4.1 Barcodes

For the prototype, barcodes were used to transfer an initial set of information from the hotel network to the user's phone. Unlike traditional barcodes such as EAN-13 which is widely deployed on consumer products, modern barcodes can store up to several hundreds of bytes of data. The mapping between a barcode and the stored information is described by a barcode symbology. To date, more than 1000 symbologies exist, differing e.g. in data density, maximum capacity, available readers or distribution.

QR Codes (Quick Response Codes) are two-dimensional codes released in 1994 and standardized in ISO/IEC18004 [11]. They provide sufficient capacity to encode the required data set. Due to their availability as an open standard, several SDKs exit for encoding and decoding QR Codes with acceptable performance.

In our initial implementation, the connectivity information was encoded in a 2D barcode and included a globally routable address and a unique identifier for each user device, generated and published by the Residential Control Device. This flow was later extended to retrieve only the address where the list of available home devices could be fetched. Prior to encoding, information could optionally be encrypted, i.e. requiring the user to enter an access code on the phone, to prevent it from being misused.

4.2 Implementation

The prototype architecture is shown in Figure 4. The multimedia content, made available through the Media Portal, is stored in the *Digital Media Server* (DMS). Based on Java Servlet technology [12], the *Media Portal* was implemented as the web application used by clients, such as the phone client as described below. After requesting users to login to the web site, it offers a personalized menu for the selection of media content and media renderer that shall be used to playout the selected content. In addition, playback control is provided.

We used a commercial off-the-shelf gateway in a routed setup and supporting NAT control through UPnP IGD. A Java MIDlet was implemented on the phone to decode the QR Codes. Information about available media renderers, obtained from the QR Codes, is submitted to the Media Portal when accessing the user's home page in the phone's browser. We used UPnP-compliant *Digital Media Renderers* located in the hotel network, such as a DLink DSM-320.

The *Residential Control Device* used in the hotel network for the access provision to local DMR:s has been implemented on a standard Linux PC. It performs the discovery of DMR:s in the hotel network by means of UPnP, and creates a provisioning web site. Here devices available for the hotel guest can be pre-selected, together with a mapping between DMR:s and

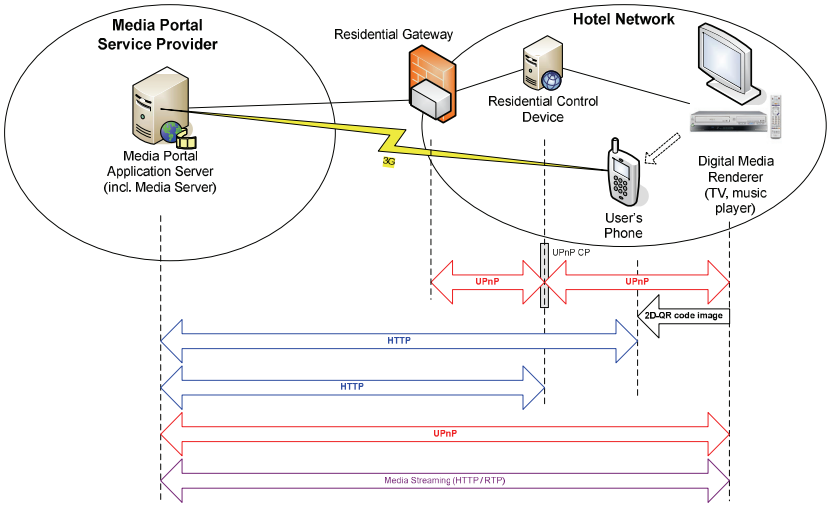


Figure 4. End-to-end prototype architecture for Online Media Portal

the guest's hotel room. The control functionality also generates the QR Code and displays it on the guest room TV via UPnP Audio-Video (AV) service.

4.3 Lessons learned

To start with, the demonstrator worked as expected and the Online Media Portal could indeed be realized as anticipated. This proofs that our architecture concepts generally hold. However, several issues have also been identified during implementation, which we summarize in the following.

The phone QR Code reader implementation as Java MIDlet yielded poor barcode recognition performance with an average delay of 6 seconds per decoding attempt. This delay can be significantly decreased to less than 2 seconds by e.g. using Symbian-based implementations. While the Java-based reader makes it necessary to manually take a snapshot of the barcode, the Symbian application allowed continuously scanning the environment for barcodes and notifying the user or an application once a code is captured. This is due to a more flexible access to the camera. The integration of barcode recognition engines into the camera API as seen for recent Japanese mobile phones is expected to further increase recognition performance.

One disadvantage of using UPnP/DLNA appliances is that only HTTP streaming is specified as mandatory, while support for RTSP/RTP is only optional. For HTTP, standard media renderers buffer parts of the media before playback. Therefore, a delay is introduced between selecting play and the actual media playout, depending on the channel capacity and required bandwidth. Our measurements show that for example for a 128 Kbps mp3 music playout at 1 Mbps access capacity around 2 seconds are spend on buffering. At 0.2 Mbps the experienced buffer delays were already in the order of eight seconds.

The timing of setting up and releasing IMS session for remote access needs to take scalability into account. In addition to the session establishment, when device control is requested as in Figure 2, the establishment may be triggered by the startup of the Residential Control Device or by the user's sign-in/sign-out to the service portal. The former may consume unnecessary resources in the network and in the AS regardless how many home devices are accessed, and the latter requires a timer based session management, since the user may leave a session without explicitly signing out. A different approach is to use SIP MESSAGE for delivering UPnP actions to home devices, without establishing a remote access tunnel. However, the user may experience longer latencies before the action is executed on the target device, because the SIP MESSAGE is routed via the CSCF.

Support for non-UPnP devices (e.g. Apple Bonjour or SIP devices) requires a new functional entity in our architecture. Currently the service provider in the AS is tightly coupled to the UPnP protocol used by the home device where the service is delivered. The new entity would be expected to provide a common interface to control home devices in order to keep service provider's logic independent from the protocol or standard used by the home device.

In case the Residential Control Device is not collocated with the Residential Gateway, the proposed solution requires support of UPnP IGD in order to establish port mappings. Since malicious software exists that exploits loopholes in IGD, several gateway vendors and operators mandate UPnP IGD to be disabled by default. The gateway working committee of the UPnP Forum is already addressing these security issues. It is likely that next IGD specification will require use of a default authentication and authorization mechanism. Alternatives to IGD are for further study.

5. Standardization

Work in several standardization bodies is ongoing that addresses most parts of our proposed architecture.

The Home Gateway Initiative (HGI, [12]) is in the process of specifying the next generation requirements on residential gateways, including an IMS Proxy function that terminates the operator NGN signaling and translates it to home network internal SIP/UPnP signaling. Our proposed HIGA architecture matches closely with HGI specifications.

ETSI TISPAN [13] is working on similar features within WG5 (Home Networks) on the Consumer Network Gateway (CNG) specifications, in close collaboration with HGI.

The UPnP Forum has been since 2006 working on specifying an architecture for remote access, basically extending the UPnP network to include remote clients via a VPN tunnel. Similarly, DLNA has recently started a task force on remote access. Both solutions can be applied to perform end-to-end signaling between a remote client (e.g. mobile phone), and UPnP/DLNA devices residing in a remote environment.

Finally, the Open Mobile Alliance (OMA) [4] is working on standardizing profiles for proximity solutions such as NFC and barcodes.

6. Conclusions & outlook

We have shown how NGN technologies, UPnP/ DLNA and new proximity solutions can be applied to separate service control from delivery. Using standard technologies available today, services can be delivered to consumer devices in broadband-connected local network islands, with the user's phone staying in control of service access and delivery. This allows operators to further adopt the role of a service and trust broker for users. This role may be further extended by offering hosting services for the access control logic to providers of network islands, such as hotels, hotspots or conference venues.

Besides architectural details and signaling flows, we have described a proof-of-concept implementation of our solution that showed that the proposed mechanisms work as expected.

We also pointed to some areas for improvements, for example the fact that proximity technologies are not yet properly integrated into the mobile phone software stack. Our prototype implementation also did not yet make use of the in-built QoS policy management and control mechanisms of IMS/NGN. For improved experience and reduced end-to-end delays, the user should be able to decide if content is streamed over a reserved channel or where best-effort delivery is sufficient, which may correspond to different charging schemes. Similarly, the use of RTP is expected to yield quality gains over HTTP streaming.

References

[1] Digital Living Network Alliance: Interoperability Guidelines v1.5, March 2006.

[2] UPnP Device Architecture 1.0, July 2006.

[3] 3GPP – http://www.3gpp.org

[4] OMA – http://www.openmobilealliance.org

[5] Open IPTV – http://www.openiptvforum.org

[6] Stefan Hoferer: Design, analysis, and prototyping of an architecture for bootstrapping of user, device, and service relationships in heterogeneous networks, diploma thesis, RWTH Aachen, March 2008.

[7] Torbjörn Cagenius, Andreas Fasbender, Luis Barriga: An IMS Gateway for Service Convergence in Connected Homes, 45th FITCE congress, Athens, August 2006.

[8] Generic Authentication Architecture and Generic Bootstrapping Architecture, 3GPP TR 33.220.

[9] Andreas Häber, Martin Gerdes, Frank Reichert, Ram Kumar: Remote Service Usage through SIP with Multimedia Access as Use Case, PIMRC 2007, Athens, September 2007.

[10] Internet Gateway Device (IGD) v1.0, November 2001.

[11] ISO/IEC. Information technology: Automatic identification and data capture techniques – QR Code barcode symbology specification, ISO/IEC 18004, August 2006.

[12] J2EE Java Servlet Technology – http://java.sun.com/products/servlet

[13] Home Gateway Initiative – http://homegatewayinitiative.org

[14] ETSI TISPAN – http://www.etsi.org/tispan

Media Delivery to Remote Renderers Controlled by the Mobile Phone

Andreas Fasbender[1], Martin Gerdes[1],
Takeshi Matsumura[2]
Ericsson Research
[1]Aachen, Germany / [2]Tokyo, Japan

Andreas Häber, Frank Reichert
Faculty of Engineering and Science
University of Agder
Grimstad, Norway

***Abstract*— In today's content delivery solutions, service delivery and control are still tightly coupled, a service typically being delivered to the same device that controls the session. We present a solution that was designed with the goal to decouple service control and delivery. Using our approach, multimedia streaming services can be delivered to off-the-shelf DLNA devices in visited networks. The service provider receives information about the remote media player and access environment via a mobile phone. Proximity technologies (e.g. barcodes, NFC) of the control device are used for the exchange of required credentials. This paper describes a typical scenario and our prototype implementation.**

Keywords- NGN, DLNA, remote media access, mobile phone control, QR-codes

I. INTRODUCTION

Entertainment devices, such as music players and cameras, nowadays come with networking capabilities that enable them to upload, download, and render media from other devices. The Digital Living Network Alliance (DLNA) [1], based on the Universal Plug and Play (UPnP) standards family [2], is now widely accepted in the consumer electronics industry. In parallel, fueled by rapidly growing mobile and fixed broadband penetration, consumers are increasingly using online media download and streaming services, such as music portals.

Network operators, on the other hand, have started to prepare for an increasing media mix by rolling out Next-Generation Network (NGN) infrastructures [3] and services based on IP Multimedia Subsystem (IMS) for service control and IP transport, based on standards of the 3rd Generation Partnership Project (3GPP) [4].

We envision an architecture that combines the benefits of operator-guaranteed trust, security, charging and quality of service (QoS) based on NGN technologies with the consumer electronics (CE) industry perspective of launching attractive end user devices.

II. DEMONSTRATION SCENARIO

A. Overview

The baseline scenario for our demonstration is depicted in Fig. 1 and described in the following:

Caroline is on a business trip, visiting a conference. In the taxi to the hotel she accesses the media portal of her service provider to listen to music with her mobile phone. After she has checked in, she decides to move the session to her hotel room, to enjoy the better sound quality of the hotel stereo system. Using her phone she connects to available media devices in the hotel room, with the stereo system and a TV set among them. Caroline selects the stereo as target device and her songs are immediately played out. With her phone, she can flip through her personalized playlist using trickplay controls such as pause, restart or forward.

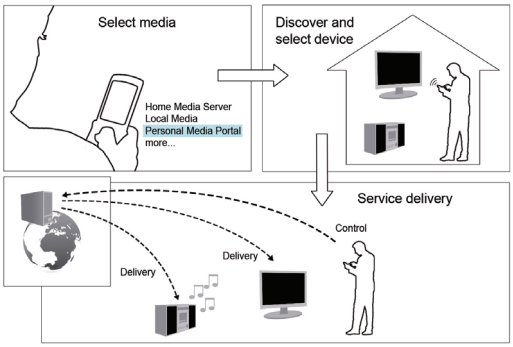


Fig. 1: Remote media access scenario

B. Relationship bootstrapping via barcodes

After the guest has checked in at the hotel reception, the residential control device creates a welcome message together with a 2D barcode (in our implementation we used the widely deployed QR code symbology). This barcode contains connectivity information about devices in the guest's room. The welcome message can for example be shown on the guest's TV or it can be handed out on paper. Prior to encoding, the barcode information can be optionally encrypted, i.e. requiring the user to enter an access code on the phone, to prevent malicious use of remote resources.

In their room, guests can use a media access client application on their phone to capture the barcode with the phone's camera. Thereafter, the client application launches the media portal in the web browser. During this process information from the barcode is passed to the media portal. The media portal offers personalized media content and a list of available media players. Guests can control the playout with their mobile phone through the media portal.

III. PROTOTYPE IMPLEMENTATION

Our demonstrator was developed in collaboration of Ericsson Research, University of Agder and RWTH Aachen. It is a simplification of our envisioned target architecture [7] and focuses on the separation of control signaling and media delivery to rendering devices.

In Fig. 2 the main demonstration flow is shown. The residential gateway connects the hotel network to the Internet. In our prototype we use a commercial off-the-shelf device supporting NAT control through the UPnP Internet Gateway Device (IGD) profile [5]. As the hotel network uses private IP addresses the internal entities cannot be reached from external hosts, such as the media portal. Furthermore, due to UPnP restrictions the Digital Media Renderer (DMR) can not locate media on external hosts. Finally, a major design goal was to support standard feature phones without mandating availability of local WiFi connectivity. We therefore assume only a cellular communication interface that allows accessing the media portal over the air.

The residential control device is used in the hotel network to configure access to local target devices, such as a DMR. It is implemented as a standard Java application. It performs the discovery of DMR:s in the hotel network by means of UPnP and provides a web console for administration. The list of devices made available to the hotel guest can be pre-configured, together with a mapping between devices and the guest's hotel room. The control functionality then generates the personalized QR code and displays it on the guest room TV via the UPnP Audio/Video service. Encoded in this QR code is a room-specific URI that addresses the residential control device instance from where connectivity information and unique identifier of each available device in the hotel room can be retrieved.

On the mobile phone a Java MIDlet was implemented to capture and decode the barcodes. Information about available media renderers, as obtained from the QR codes, is submitted to the media portal when accessing the user's home page. The prototype supports UPnP compliant DMR:s located in the hotel network, such as a Zyxel DMA-1000.

The media portal offers a personalized menu for the content and media renderer selection. According to the user's trickplay requests on the media control page, UPnP control actions are delivered to the target device through a dedicated port mapping in the residential gateway.

IV. CONCLUSIONS

Using standard technologies available today, services can be delivered to off-the-shelf consumer devices in broadband-connected local network islands, with a user's phone staying in control of service access and delivery.

We have presented a proof-of-concept implementation that shows that the proposed mechanisms work as expected. Moreover, we have discovered some areas for improvement of the perceived usability that will be considered for the next evolution step of the prototype.

One area for improvement is system responsiveness. The recognition of QR codes by the camera and the decoding would be faster if implemented as a native application, as is the case with in most Japanese phones today. Latencies between issuing play commands and actual media playout can be improved by using RTP/RTSP for media delivery and by negotiating appropriate QoS settings for the signaling of UPnP actions. Here, the use of IMS would yield clear advantages.

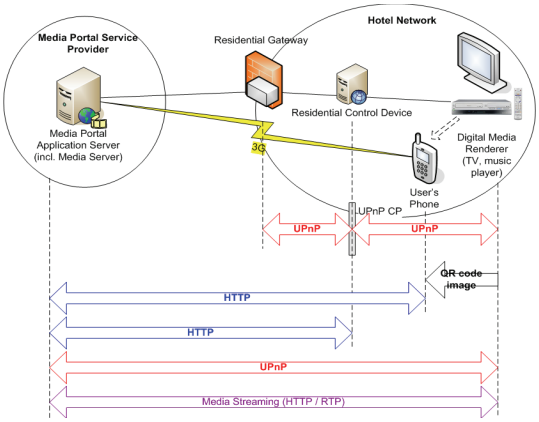


Fig. 2: High-level system architecture and signalling flows.

In deployments where the residential RCD is not co-located with the gateway, the proposed solution requires support of UPnP IGD in order to establish port mappings. However, several gateway vendors and operators mandate i to be disabled by default for security reasons. The next version of the UPnP IGD specifications will provide secure default authentication and authorization mechanisms.

In line with our envisioned architecture, we are currently working on an enhanced version of the prototype, as outlined in [7]. This enhanced version utilizes IMS Presence Services [6] for remote service discovery and contains improvements addressing the aforementioned limitations.

REFERENCES

[1] "Digital Living Network Alliance (DLNA): Enjoy your music, photos and videos, anywhere anytime", http://www.dlna.org/.

[2] "UPnP Device Architecture", UPnP Forum specifications, 2003. http://www.upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0.pdf.

[3] M. Carugi, B. Hirschman, A. Narita, "Introduction to the ITU-T NGN focus group release 1: Target environment, services, and capabilities", IEEE Communications Magazine, vol. 43, pp. 42-48, 2005.

[4] "Third Generation Partnership Project (3GPP)", http://www.3gpp.org/.

[5] "Internet Gateway Device", P. Iyer and U. Warrier, Eds., UPnP Forum specifications, 2001, http://www.upnp.org/standardizeddcps/igd.asp.

[6] "Presence service; Architecture and functional description; Stage 2", 3GPP TS 23.141, http://www.3gpp.org/ftp/Specs/html-info/23141.htm.

[7] A. Fasbender et al., "Phone-controlled Delivery of NGN Services into Residential Environments," Proc. 2nd IEEE Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST 2008), Cardiff, Wales, 2008.

DELIVERING SERVICES TO RESIDENTIAL APPLIANCES BY UTILIZING REMOTE RESOURCE AWARENESS

Andreas Häber
University of Agder
Grimstad, Norway
andreas.haber@uia.no

Martin Gerdes
Ericsson GmbH
Aachen, Germany
martin.gerdes@ericsson.com

Frank Reichert
University of Agder
Grimstad, Norway
frank.reichert@uia.no

Andreas Fasbender
Ericsson GmbH
Aachen, Germany
andreas.fasbender@ericsson.com

Ram Kumar
University of Agder
Grimstad, Norway
ram.kumar@uia.no

***Abstract*—*Service providers are nowadays offering a variety of services, and in particular multimedia content delivery. Besides, consumer appliances are increasingly becoming digitalized including support for communication networks. However, it is difficult, and in many cases impossible, to use these services with standard consumer appliances, such as TV and media player devices. Rather, usage is often restricted so that they can only be accessed through web browsers from PCs, mobile phones and similar terminals. This is unfortunate, because dedicated consumer appliances are often better suited to handle the content and thereby give consumers a better experience. Within this paper, three design approaches that support such services are described and compared, along with a prototype that shows this concept.*

***Index Terms*—Residential network, IMS, SIP, presence, multimedia systems, service discovery**

I. INTRODUCTION

SERVICE PROVIDERS are nowadays offering a variety of services, and in particular multimedia content delivery. For example, Internet TV, Video on Demand (VoD) and real-time streaming of events, such as sport and concerts, can be enjoyed from the Internet. Besides, consumer appliances are increasingly becoming digitalized including support for communication networks. In [1] a use case for a personal media portal is described that shows how these two aspects can be combined. However, it is difficult, and in many cases impossible, to realize this use case by using these external services together with standard consumer appliances, such as TV and media player devices. Rather, usage is often restricted so that they can only be accessed through web browsers from PCs, mobile phones and similar terminals. This is unfortunate, because dedicated consumer appliances are often better suited to handle the content and thereby give consumers a better experience.

Within residential networks service discovery protocols (SDPs), such as Universal Plug & Play (UPnP) [2] and Apple Bonjour [3], are used to be aware and utilize resources and services, including those provided by consumer appliances. Through these SDPs, applications can be aware of and utilize services in the network. For service providers to be able to deliver content to consumer appliances they also need to be aware and have access to these appliances. However, because the SDPs have been designed to work only in local networks, service providers cannot directly utilize them. This limitation is, of course, good when it comes to illegitimate access to residential services, but it also denies service providers to discover and communicate with these residential services. Therefore, a solution to enable this legitimate remote access is necessary.

Remote service discovery, enabled with service presence, promises such a solution to this problem. Residential networks publish service presence information (including service status and a service description) to a central location. Authorized users, such as service providers, can utilize this presence information to access and use remote services [4]. Thereby, richer applications can be created that allow users to utilize their residential services together with their IM identity.

However, there are many different possible solutions based on these enablers. In this paper, we describe different solution approaches (Section IV) and compare them against each other (Section V).

II. A LOOK AT CURRENT SERVICE DELIVERY

Today, many service providers utilize web applications for user access to their services. With the advances of web technology, such as Ajax, rich applications can be created [5]. For online video services, such as TV2 Sumo [6], usually a web page is presented to users where the content is selected. Then, the content is shown in an embedded media player, where also the trickplay options, such as playback, are controlled.

However, the usability of the service is limited because it can only be consumed within a media player embedded in the web browser instead of an external target device. Notably, the media player selection step, part of the use case described in [1], is excluded. One workaround for this coupling is to fetch the Uniform Resource Identifier (URI) [7] from the information returned by the service and transfer it, using an ad-hoc mechanism, to the target device. Nevertheless, because many systems enforce access control and charging on content, such a workaround has limited effect. Additionally, such workarounds severely cripples the user experience.

Furthermore, it is also desirable to use different data channels for controlling the session and retrieving the content. For example, it should be possible to control the session using a mobile phone with a cellular link, while the media is received using a fixed broadband connection. This way, the

user's identity on the mobile phone will be used for authentication, authorization and accounting (AAA) and session control, and it is freed from handling the media and forwarding it to the target device.

III. Common Enablers for Residential Service Delivery

Although there are different possible solutions for the delivery of remote services into residential appliances (see section IV), they share a set of underlying requirements. These requirements are fulfilled by some common enablers that will be part of any of the possible solutions. The enablers are introduced and explained within this section.

A. Signaling protocol for converged services

As implied in section II above, currently online services are delivered through web applications based on the Hypertext Transport Protocol (HTTP) [8]. This protocol has shown to be a great success and is widely adopted and used. However, the lack of a signaling plane becomes problematic when, for example, the service (i.e. the service data, e.g. media data in case of media services) is supposed to be transmitted to a different device than the one that has actually initiated the service and requested the data.

The Session Initiation Protocol (SIP) [9] is an application layer signaling protocol for managing sessions between multiple participants. Thereby, the service session can be controlled using SIP, while at the same time the media can be transported using HTTP.

B. Identity Management and the Generic Authentication Architecture (GAA)

Identity management is an important piece of next-generation networks that will be explained briefly in the following. The IMS operator deploys a Bootstrapping Server Function (BSF), and the application server (AS) provides as Network Application Function (NAF). If a User Equipment (UE) requests a service from the AS for the first time, the AS will demand that the UE must be authenticated using GBA. Thereafter, the UE and the BSF mutually authenticate using a shared secret. As a result, a pair of session keys is generated by the BSF, and one of the keys is delivered to the UE. The UE responds back to the AS with the received session key, where after the AS requests the BSF to authenticate the user by providing the session key. The BSF returns the authentication result and finally the AS approves that the UE is authenticated. Besides high security, this process has the advantage that it can be completed without the user having to type in a password.

C. Remote Service Awareness

The underlying method to make application servers aware of remote residential devices and their services is based on an extension to the IETF's Presence model [10] termed *Service Presence*. This extension is based on the transport and provision of service related information through SIP networks, such as IMS. Information about devices and services discovered in one network environment are transported via the common communication infrastructure to another, remote network environment. For that it uses an extended presence data model including generic descriptions for services, including a service access interface description, parameters, and service state values. This allows a presence watcher in the application server to be aware of the remote devices and services.

The architecture that realizes this Service Presence concept includes four logical nodes, as explained in the following.

1) Service Presentity: A residential appliance like a Digital Media Renderer uses a Service Discovery Protocol (SDP) mechanism (for example, UPnP) to publish its own services within the residential environment.

2) Service controller: The Service Controller is an entity that discovers services, using the same SDP mechanism as the service providers, and controls or uses them.

3) Service Discovery Gateway (SDG): The SDG is responsible for the discovery of services in its vicinity (via SDP) and for the publication of their presence to make them available for remote access.

Furthermore, it can subscribe to receive service presence information from other SDGs.

4) Service Presence Server (SPS): This node is introduced in the core network to support the SDGs by collecting service presence information from them. This offloads the SDGs with the task of notifying all presence watchers.

IV. Design Approaches

In the following, three design approaches are presented that support using residential services as part of a session. These approaches differ in the point of time when the selection of the target media player happens compared to the content selection in the service, provided as web application: before the content selection, after the content selection, and an integrated selection (i.e. both happen in the service's web application). As will be shown, this opens up for different roles for the nodes involved in the session.

Common for all solutions is that GAA (see III.B) is used to handle authentication and authorization between the service and the user when necessary. In addition, only requests are shown to simplify the sequences.

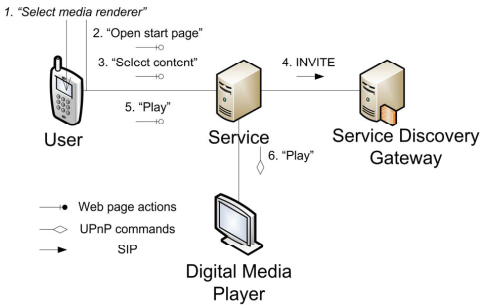


Fig. 1 Pre-selection solution.

A. *Pre-selection of the target device*

In this approach, the user selects the target device before contacting the service provider. This minimizes the changes required to the web application, and, in particular, to the user interface because it does not need to be adapted to media player selection. Therefore, it can be used to add ad-hoc support for device selection. For example, an application for device selection can run on the user's control terminal, such as a Java MIDlet [11] running on a mobile phone.

After target device selection, this application is responsible to contact the user's service provider and transmit information about the selected device. This can simply be achieved by starting a web browser that shows the service's start page.

Next, in the service provider's web application the user selects the desired content. After selection, the web application establishes a session with the SDG for controlling the target device. Moreover, in this session the web application can control the target device using its control protocol, such as SOAP [12] if it is an UPnP-device. Playback can then be controlled using the service's web application. Alternatively, the playback control can be transferred to the user's control terminal. However, such control transfer is out of scope for this paper.

B. *Post-selection of the target device*

This solution takes the opposite approach of the pre-selection solution. First, the user starts by opening the web application of the service, as described in section II. However, instead of the media being rendered in the user's web browser, the service sends a SIP INVITE request to the user to establish a session that will include the media flow. This INVITE request can be handled in two ways by the user's application:

1) **Gateway as terminal**
   It can terminate the SIP dialog by acting as a gateway and communicate with the target device by itself. For example, for this communication UPnP can be used. This approach is illustrated in Fig. 2.

2) **Refer to a gateway (i.e., a SDG)**
   Based on the user's selection the application may reply with a REFER request that refers the service to an SDG, along with information on which device to use. This SDG should handle the request by setting up remote connectivity to the device so that the service can communicate with it.
   It should be noted that the user could refer to an SPS instead, because that node will redirect the request to the appropriate SDG. The communication flow is shown in **Error! Reference source not found.**.

A crucial difference between these two options is that the former option requires the user's device to be capable of controlling the target device, while the latter option only requires SIP to communicate with the Core Network.

Moreover, as the former approach requires the terminal to communicate with the device itself additional resources are required. Nowadays low-end phones are not equipped with such resources, and therefore we do not consider this option suitable for low-end phones.

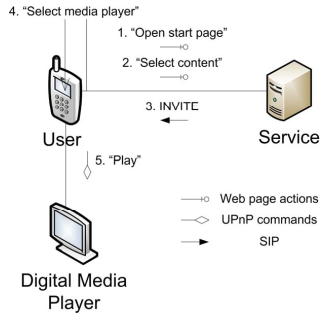


Fig. 2 Post-selection solution, alternative 1.

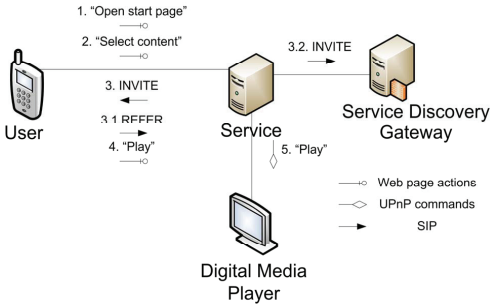


Fig. 3 Post-selection solution, alternative 2.

C. *Integrated selection of the target device*

In this solution, the selection of the target device is integrated with the rest of the user interface. Therefore, after authentication and authorization of the user, the online service provider will acquire the available target devices using remote service discovery.

As illustrated in Fig. 4, the user starts by connecting to an online service using web-based technologies (step 1). Furthermore, the online service retrieves service presence information for relevant services from SPS (step 2). This step can be achieved in different ways, such as querying it for all the user's available media renderers supporting a suitable format. Next, the user selects the content and the media player(s) to be used for this session. Furthermore, included in the service presence information received from SPS is the address of the remote service's service discovery gateway (SDG). This address is used to establish a remote service usage session between the online service and the remote device(s) selected.

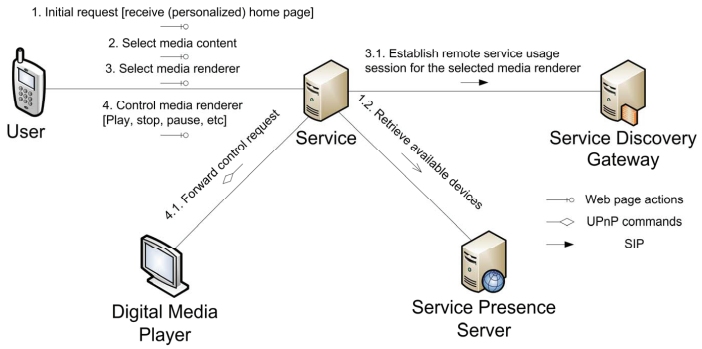


Fig. 4 In-selection approach.

V. Comparison of the Design Approaches

The approaches proposed in Section IV above have positive and negative sides, and those will be described and compared in this section. A summary of the comparison is given in Table 1.

A. Session control terminal involvement

Criteria description: Whether the session control terminal must handle the content/media or not.

Only alternative one of the post-selection approach requires that the session control node must handle the media.

B. Integration with existing systems

Criteria description: The amount of work expected to integrate a solution with an existing system.

Alternative one of the post-selection solution is conceived to be easiest to integrate with existing systems, because it leaves most of the work to the client. All the other solutions require the service to extend the functionality to be able to control the media player, such as UPnP control point logic.

It is expected that both the pre-selection and the alternative two of the post-selection solution will require the same amount of integration work.

Nonetheless, the in-selection approach is conceived to be most difficult to be integrated with existing systems because it requires changes in the user interface and that the signaling flow changes as it must now contact more nodes.

C. Change content selection without restarting the session control

Criteria description: Whether the user may change to use different content without having to restart from the beginning of the flow.

With alternative one of the post-selection it is not feasible to change the content selection, because the service gives away control of the media flow to the user. For the other approaches it is possible to change the content selection as the service will be able to control the media flow.

Table 1 Comparison of the different solutions.

Criteria	*Pre-selection*	*Post-selection, alt 1*	*Post-selection, alt 2*	*Integrated selection*
Session control terminal involvement	No	Yes	No	No
Integration with existing systems	Middle	Easiest	Middle	Most difficult
Change content selection without restarting the session control	Yes	No	Yes	Yes
Transfer ongoing sessions	No	No	Yes	Yes
Hardware requirements	Depends	High-end	Low-end	Low-end

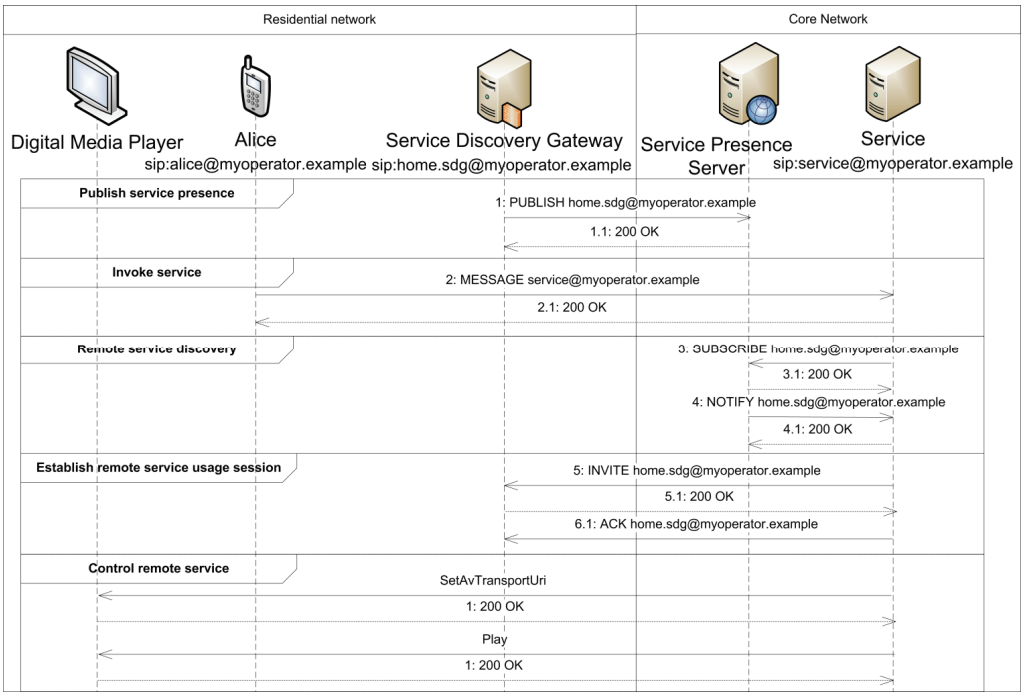


Fig. 5 Prototype signaling flow.

D. Transfer ongoing sessions

Criteria description: Whether it will be feasible to transfer an ongoing session to a different target device.

Both the pre-selection solution and alternative one of the post-selection solution does not support this criterion. The session must be restarted and then select the new target device.

E. Hardware requirements

Criteria description: Whether the solution requires high-end, mid-end or low-end terminals.

Alternative one of the post-selection solution requires most hardware capabilities, as it must handle the media and be able to directly control the media player device.

Depending on which mechanism the pre-selection solution uses for target device selection it may also require high-end hardware, such as Wi-Fi support for discovery.

VI. Service Delivery Prototype

For a proof of concept, a simple prototype system based on the post-selection approach, alternative 2, has been developed. This prototype shows whether it is possible to let an external service control a residential network, based on user request. As shown in Fig. 5, the prototype includes a service that will simply display a picture on a media player of the client when requested.

In the following, first an overview of the components are given followed by an outline of the signaling flow, and finally core network configuration is described.

A. Components

1) *Digital Media Player:* A UPnP Media Renderer device.
2) *Service Presence Server:* A presence server that supports the service presence extensions.
3) *Service Discovery Gateway (SDG):* This implementation of an SDG discovers local UPnP devices and publishes their presence to the Service Presence Server. In addition, it supports remote usage of these devices by adding port mappings to control requested devices.
4) *Alice (client):* This client sends a MESSAGE-request that includes the address of its SDG. In our implementation the "Test Agent " tool from the Ericsson Service Development Studio (SDS) was used.
Here, our client-side implementation differs from how it is described above, where it is a converged application instead. Another difference is that the service does not send a REFER-request but rather uses the SDG-address included in the MESSAGE-request sent by the client. This also limits the selection choice of the client, as it does not include any specific device to be used.
5) Display Image Service: Processes MESSAGE-requests and displays a picture on the first available Digital Media Player of the client.

B. Signaling flow

As a pre-requisite, there must be a Digital Media Player

located in the residential network. The SDG discovers this device and publishes its presence status with the Service Presence Server (step 1 in Fig. 5).

To invoke the service the client sends a MESSAGE-request with a body set to the IP Multimedia Public Identity (IMPU) of the SDG (step 2 in Fig. 5). When the service receives this request it first retrieves the available media players from the service presence server (step 3 and 4 in Fig. 5). Finally, it establishes a remote service usage session with the SDG (step 5 in Fig. 5) that it uses to control the device to display a picture in the final phase.

C. *Core network configuration*

The Picture display service is provisioned with a Public Service Identifier (PSI), which, amongst other uses, the client uses to send a MESSAGE request to. In addition, an initial Filter Criteria (iFC) is created for the service presence service so that PUBLISH and SUBSCRIBE requests are routed to it.

VII. Conclusion

In this paper, three design approaches for delivering services to residential appliances have been described and compared. The prototype shows that the concept of using remote resource awareness to deliver services to residential appliances works as expected.

As the comparison shows, the three design approaches fit different scenarios, for example whether it will be a completely new service or if support will be integrated with an existing one. Nonetheless, in the future it should be investigated the scalability requirements of these design approaches, as that is very important when such services will be used by many concurrent clients.

A. *Future work*

The prototype is too simple to cover the whole end-user experience promised in this solution. For example, there is no good media selection and session control implemented.

Furthermore, there are both security and privacy aspects that needs to be investigated further for this work. As described in section IV GAA can provide authentication and authorization. However, how to ensure the privacy of the users, secure transfer of media, transfer of session control, and more are left for future investigation.

References

[1] F. Reichert, A. Häber, M. Gerdes, A. Fasbender, and G. Loudon, "“Sven and the Media Portal” - A Nomadic Use Case for the Extended Home," in *15th IST Mobile & Wireless Communication Summit* Myconos, Greece, 2006. http://mobilesummit2006.org/

[2] "UPnP Device Architecture," UPnP Forum, 2003. http://www.upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0.pdf

[3] Apple, "Bonjour." http://developer.apple.com/opensource/internet/bonjour.html

[4] A. Häber, M. Gerdes, F. Reichert, R. Kumar, and A. Fasbender, "Remote Service Usage through SIP with Multimedia Access as an Use Case," in *18th IEEE Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2007)*, Athens, Greece, 2007.

[5] L. D. Paulson, "Building rich web applications with Ajax," *Computer,* vol. 38, pp. 14-17, 2005.

[6] TV2, "TV 2 Sumo," 2008. http://webtv.tv2.no/webtv/

[7] T. B. Lee, R.Fielding, and L. M. L. rfc, "Uniform Resource Identifiers (URI): Generic Syntax," 1998.

[8] R.Fielding, J.Gettys, J.Mogul, H.Frystyk, L.Masinter, P.Leach, and T. B.-L. L. rfc, "Hypertext Transfer Protocol -- HTTP/1.1," 1999.

[9] J.Rosenberg, H.Schulzrinne, G.Camarillo, A.Johnston, J.Peterson, R.Sparks, M.Handley, and E. Schooler, *SIP: Session Initiation Protocol*, IETF RFC 3261, http://www.rfc-editor.org/rfc/rfc3261.txt

[10] M. Day, J. Rosenberg, and H. Sugano, *A Model for Presence and Instant Messaging*, IETF RFC 2778, http://www.rfc-editor.org/rfc/rfc2778.txt

[11] JSR118 Expert Group. *JSR118: Mobile Information Device Profile 2.1*, J. Warden, Ed., June 2, 2006. http://jcp.org/en/jsr/detail?id_118.

[12] M. Nilo, "SOAP Version 1.2 Part 0: Primer," 2001.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 July 2008 (10.07.2008)

PCT

(10) International Publication Number
WO 2008/082346 A1

(51) International Patent Classification:
H04L 29/06 (2006.01) *H04L 12/66* (2006.01)
H04L 12/46 (2006.01)

(21) International Application Number:
PCT/SE2007/050740

(22) International Filing Date: 15 October 2007 (15.10.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/882,303 28 December 2006 (28.12.2006) US

(71) Applicant *(for all designated States except US)*: **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; 164 83 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **GERDES, Martin** [DE/DE]; Retzstr. 31, 52156 Monschau-rohren (DE). **FASBENDER, Andreas** [DE/DE]; Steppenbergallee 78, 52074 Aachen (DE). **HÄBER, Andreas** [NO/NO]; Storgaten 33, 4876 Grimstad (NO). **REICHERT, Frank** [DE/NO]; Vikstölen 14, 4885 Grimstad (NO).

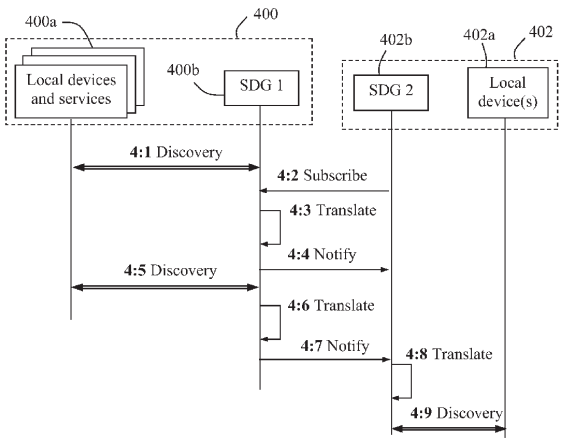
(74) Agent: **BERGENSTRÅHLE & LINDVALL AB**; HAGSTRÖM, Hans, Box 17704, 118 93 Stockholm (SE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

(54) **Title:** A METHOD AND APPARATUS FOR SERVICE DISCOVERY



(57) **Abstract:** A method and apparatus for conducting remote discovery of services across different local networks. A service discovery gateway (202b) in one local network (202) issues a request for discovery information on the services in the opposite local network (200), embedded in a presence subscribe message over an IMS network (204). Discovery information is then received in a generic format embedded in a presence notify message over the IMS network. The received discovery information has been collected by a service discovery gateway in the first local network using a local service discovery protocol in the first local network. The received discovery information is announced to devices in the second local network, using a local service discovery protocol valid within the second local network. Thereby, local services can be provided and utilized across different local networks, even when different device-specific service discovery protocols are used within the local networks.

WO 2008/082346 A1

A METHOD AND APPARATUS FOR SERVICE DISCOVERY.

TECHNICAL FIELD

The present invention relates generally to a method and apparatus for enabling remote discovery of services and communication devices across different local networks, to enable communication of media with a device in one local network from a device in another opposite local network.

BACKGROUND

A multitude of different types of communication terminals, sometimes referred to simply as "devices", have been developed for packet-based multimedia communication using IP (Internet Protocol). Multimedia services typically entail transmission of media in different formats and combinations over IP networks. For example, an IP-enabled mobile terminal may exchange media such as visual and/or audio information with another IP-enabled terminal, or may download media from a content server over the Internet.

A network architecture called "IP Multimedia Subsystem" (IMS) has been developed by the 3rd Generation Partnership Project (3GPP) as a platform for handling and controlling multimedia services and sessions, commonly referred to as the IMS network. Thus, an IMS network can be deployed to initiate and control multimedia sessions for IMS-enabled terminals connected to various access networks, regardless of the access technology used. Although conceived primarily to enable multimedia services for mobile IP terminals, the IMS concept can also be used for fixed IP terminals.

Multimedia sessions are handled by specific session control nodes in the IMS network, e.g. the nodes P-CSCF (Proxy Call Session Control Function), S-CSCF (Serving Call Session Control Function), and I-CSCF (Interrogating Call Session Control Function). Further, a database node HSS (Home Subscriber Server) is used in the IMS network for storing subscriber and authentication data. The IMS network may also include various application servers for providing different multimedia services, such as presence services where terminal users can subscribe to various published information on other users.

According to the IMS platform, the control protocol called "SIP" (Session Initiation Protocol) is utilised to initiate, operate and terminate multimedia sessions. Standard SIP messages can thus be used by IP terminals or devices for establishing multimedia sessions, such as the session initiating message "SIP invite" and the common response message "SIP 200 OK".

In SIP, the "Session Description Protocol" can also be used, embedded as a self-contained body within SIP messages, to specify different communication parameters needed for a forthcoming multimedia session. This protocol is generally used to provide necessary information in session setup procedures, e.g. device capabilities, media properties, currently used IP addresses, etc., as is well-known in the art.

It is desirable to generally provide IMS-based services also for devices in a limited IP based local or private network such as a residential or office network, also referred to as a LAN (Local Area Network) or PAN (Personal Area Network). In this description, the generic term "local network" is used to represent any such networks,

and the term "device" is used to represent any terminal capable of IP communication within the local network. In such local networks, a local IP address is allocated to each device for communication within the network which, however, cannot be used for routing messages and data outside that network.

The devices in a local network may include fixed and wireless telephones, computers, media players, servers and television boxes (the latter often referred to as a Set Top Box, STB). In order to provide IMS services to devices in the local network, a multimedia gateway called "Home IMS Gateway, HIGA" has been defined that can emulate an IMS terminal from the local network towards the IMS network, to access IMS services on behalf of any device in the local network.

UPnP (Universal Plug-and-Play) is an architecture developed in a multi-vendor collaboration called the UPnP Forum, for establishing standard device protocols for the communication between different IP devices in a local network using different access technologies, operating systems, programming languages, format standards and communication protocols. UPnP provides standardised methods to describe and exchange device profiles that include capabilities, requirements and available services in the devices.

UPnP also supports a process called "discovery" (or "pairing") in which a device can join a local network, obtain a local IP address, announce its name and IP address, and exchange capabilities and services with other devices within the network. In the following description, the term "discovery information" represents any such information such as name, identity, local IP address, URI (Universal Resource

Identifier) for stored media content, device capabilities and available services, communicated between the devices during a discovery process. The discovery process can also be conducted within a temporarily formed ad-hoc network, e.g. using Bluetooth communication.

For Bluetooth, a Service Discovery Protocol (SDP) has been standardised for finding devices and their services in the discovery process. The device capabilities and available services can be specified in a Service Discovery Application Profile (SDAP), as utilised by the SDP. For networks using other communication protocols, such as ZigBee and IrDA (Infrared Data Association), similar device profiles and service discovery mechanisms have been defined.

DLNA (Digital Living Network Alliance) is a standardisation organisation that develops interworking guidelines for acquiring, storing and accessing digital media content from devices in a local network. The UPnP protocol is utilised by DLNA as an underlying protocol for communication between devices within local networks.

An architecture for enabling remote access will also be defined, where remote "UPnP devices" located outside the local network can communicate media with devices within the network. In WO 2006/079891 (Nokia), a solution is described for setting up a VPN (Virtual Private Network) tunnel as a data/media transport channel for such remote UPnP access, e.g. using IPSec (IP Security). However, this solution requires the use of IP address resolution and DNS (Domain Name server) technology, as well as access to a dynamic DNS client in the private network.

In Fig. 1, a local network 100 is shown with different devices in a family residence or an office. As shown here, these devices include a wireless telephone, a

fixed telephone, a TV apparatus, a laptop computer, and a media server. The network 100 also includes a conventional gateway 102 connected to an external access network 104 to provide a communication link to other networks for the devices, referred to as a "residential gateway RGW". The RGW 104 typically includes a NAT (Network Address Translation) function and a local DHCP (Dynamic Host Configuration Protocol) server allocating local IP addresses to the devices, as is well-known in the art.

The local network 100 further includes a HIGA 106 providing a connection to an IMS network 108 in which an HSS 110 is shown. The HIGA 106 has suitable interfaces towards the different devices in network 100, using device-adapted protocols. The HIGA 106 may be integrated in the RGW 102, but logically it will be considered as an individual functional unit in this description.

The HIGA 106 holds IMS identity information 112 associated with IMS subscriptions and user/service profiles, which can be used for accessing the IMS network 108 where corresponding subscriber information 114 is stored in the HSS node 110. Accordingly, a user can log on to the IMS network from a specific used device in the local network 100 by means of HIGA 106, and the local IP address of that used device will then be associated with the user's profile. In WO 2006/045706 (Ericsson) it is described how devices in a local network can obtain IMS services by means of a HIGA.

When HIGA 106 receives a request for a multimedia service from a device in network 100 using a device-specific interface/protocol, HIGA 106 translates the service request into a valid IMS request (e.g. SIP invite) on behalf of the device, to set up a session for the device by communicating suitable SIP messages with the IMS network 108, accordingly.

In a similar manner, an IMS session can be set up by HIGA 106 for an incoming request for communication with a device in network 100, by using an IMS identity 112 associated with the device. In either case, communicated media is routed during the session from or to the device over the RGW 102 and the access network 104, as indicated by two-way arrows.

Fig. 1 further illustrates that a local device 100a moves outside the network 100 to become a remote device 100a'. The remote device 100a' can then send a SIP invite message to the HIGA 106 over the IMS network 108 to initiate media communication with one of the remaining devices in network 100. The remote device must then have a valid IMS identity for accessing the IMS network.

In order to communicate with a device in a local network from a remote device located outside the network, the remote device must first gain knowledge of the other device, and vice versa, in a discovery process. Once a device has executed the discovery process in a local network, it has knowledge of the local IP-address, name and capabilities/services of other devices in that network. The devices can then exchange media content inside the network, but not outside since the local IP address cannot be used. Thus, should a device move out of the local network and connect to some other network, it can no longer interact with the local devices in the first network in this manner and discovery messages cannot be exchanged remotely.

Moreover, when a device belonging to a first local network moves to a second local network using an allocated local IP address for communication in the second network, it is not possible to conduct a discovery process remotely with devices in the first local network. Therefore, the remote device cannot find devices and services in the first local

network, nor announce itself and its services to the first network, when located in the second local network.

It would be complicated and difficult to realise applications on a single device that can obtain and use information on services in a remote network and also discover local services in a currently connected network, and further provide discovery information on the local services to the remote network. Today, no solution exists for discovering and utilising services across different local networks that use different service discovery protocols (SDPs), as the different SDPs are not compatible to each other. For example, a device that only supports a UPnP-based SDP for service discovery cannot utilise any Bluetooth service provided by another device, neither remotely from another local network nor locally within the same network.

For example, it would be desirable to discover services and devices in a remote network from a device (e.g. a mobile IMS phone) located in another local network, in order to utilise a service in a device (e.g. a media server) in the remote network basically in the same manner as service consumers would do within that network. It may also be desirable to provide information about services and devices discovered in the current local network to a remote local network, so that the local services can be utilized from service consumers within the remote network.

SUMMARY

It is an object of the present invention to address at least some of the problems outlined above. Further, it is an object to provide a solution for the discovery of devices and/or services across different local networks to enable

multimedia sessions. These objects and others may be obtained by providing a method and apparatus according to the independent claims attached below.

According to one aspect, a method is provided to enable remote discovery of services and devices in a first local network from a location within a second local network, as executed by a service discovery gateway in the second local network. In this method, a request is issued for discovery information on the devices in the first local network, the request being sent embedded in a presence subscribe message over an IMS network. In response to the request, discovery information is received in a generic format embedded in a presence notify message over the IMS network. The received discovery information has been collected by a service discovery gateway in the first local network in a discovery process using a local service discovery protocol valid within the first local network. The received discovery information is then announced to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

According to another aspect, a service discovery gateway is provided for conducting remote discovery of services and devices in a first local network from a location within a second local network, where the service discovery gateway is connected to the second local network. The service discovery gateway comprises a presence watcher unit adapted to issue a request for discovery information on the devices in the first local network, where the request is sent embedded in a presence subscribe message over an IMS network. The presence watcher is further adapted to receive discovery information embedded in a presence notify message

over the IMS network in response to the request. The received discovery information has been collected by a corresponding service discovery gateway in the first local network using a local service discovery protocol valid within the first local network. The service discovery gateway of the second local network further comprises an announcing unit adapted to announce the received discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

The service discovery gateway of the second local network further may comprise a translator adapted to translate the discovery information from the generic format into a local format supported by the second local network, before being announced in the second local network.

Different embodiments are possible in the method and service discovery gateway of the second local network above. For example, the discovery information may further be translated from the generic format into a local format supported by the second local network, before being announced in the second local network.

The request for discovery information may be sent to the service discovery gateway in the first local network, and the discovery information is then received as a response from that service discovery gateway. Alternatively, the request for discovery information may be sent to a presence server in the IMS network, and the discovery information is then received as a response from that presence server. In the latter case, the presence server may have received the collected discovery information in a presence publish message from the service discovery gateway in the first local network.

In practice, the second local network could be an ad hoc network and the service discovery gateway in the second local network could be implemented in a user device acting as a multimedia gateway to access the IMS network on behalf of other devices in the ad hoc network. In that case, a portable IMS gateway "PIGA" may also be implemented in the user device.

According to yet another aspect, a method is provided to enable remote discovery of services and devices in a first local network from a location within a second local network, as executed by a service discovery gateway in the first local network. In this method, discovery information of the services and devices in the first local network is collected in a discovery process using a local service discovery protocol valid within the first local network. The collected discovery information is then provided in a generic format embedded in a presence message over an IMS network, thereby enabling a service discovery gateway in the second local network to announce the discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

According to yet another aspect, a service discovery gateway is provided for enabling remote discovery of services and devices in a first local network from a location within a second local network, where the service discovery gateway is connected to the first local network. The service discovery gateway of the first local network comprises a discovery unit adapted to collect discovery information of the services and devices in the first local network in a discovery process using a local service discovery protocol valid within the first local network. The

service discovery gateway further comprises a presence presentity unit adapted to provide the collected discovery information in a generic format embedded in a presence message over an IMS network, thereby enabling a service discovery gateway in the second local network to announce the discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

The service discovery gateway of the first local network may further comprise a translator adapted to translate the discovery information obtained in a local format into the generic format, before being provided over the IMS network.

Different embodiments are possible in the method and service discovery gateway of the first local network above. For example, the presence message may be sent as a presence notify message to the service discovery gateway in the opposite second local network. Alternatively, the presence message may be sent as a presence publish message to a presence server in the IMS network.

According to yet another aspect, a presence server is provided in an IMS network for enabling remote discovery of services and devices in a first local network from a location within a second local network. The presence server comprises an event state compositor adapted to receive discovery information on devices in the first local network, in a generic format embedded in a presence publish message from a service discovery gateway connected to the first local network. The presence server further comprises a presence agent adapted to receive a request for the discovery information, embedded in a presence subscribe message, from a service discovery gateway connected to the

second local network. The presence agent is further adapted to send the discovery information embedded in a presence notify message to the service discovery gateway in the second local network in response to the request, thereby enabling the service discovery gateway in the second local network to announce the discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

Further possible features and benefits of the present invention will be explained in the detailed description below.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be explained in more detail by means of preferred embodiments and with reference to the accompanying drawings, in which:

- Fig. 1 is a schematic view illustrating a local network when a remote device accesses the network from a location outside the network, according to the prior art.
- Fig. 2 is a schematic network overview illustrating procedures for service and device discovery across two different local networks, in accordance with different embodiments.
- Fig. 3 is a flow chart illustrating a procedure for enabling remote discovery across two opposite local networks, in accordance with one embodiment.
- Fig. 4 is a signalling diagram illustrating how the inventive solution can be implemented in the peer-to-peer case, in accordance with another embodiment.

- Fig. 5 is a signalling diagram illustrating how the inventive solution can be implemented in the IMS-centric case, in accordance with yet another embodiment.
- Fig. 6 is a schematic block diagram illustrating two service discovery gateways in opposite local networks when using the peer-to-peer method for remote discovery, in accordance with yet another embodiment.
- Fig. 7 is a schematic block diagram illustrating two service discovery gateways in opposite local networks when using the IMS centric method for remote discovery, in accordance with yet another embodiment.
- Fig. 8 is a schematic block diagram illustrating a service discovery gateway, in accordance with yet another embodiment.

DETAILED DESCRIPTION

Briefly described, the present invention can be used to enable remote discovery and utilisation of services and/or devices in one local network, from a device located in another opposite local network. The two local networks may use different internal service discovery protocols to communicate discovery information within each network, where the internal service discovery protocols may well be incompatible.

In this solution, the discovery process can be conducted between a device in one network and devices in the other opposite network, where the existing communication framework for presence services in an IMS network is used to convey discovery information in a generic format as "service presence information" between the two networks.

Conventionally, presence services make information related to a specific client available to other clients.

Thus, presence data is stored in a presence server for providing such data to subscribing clients, and certain access rights can then also be enforced for different clients. The presence data of a client may typically relate to his/her status, device capabilities, geographic location, and other information such as interests, occupations, skills, characteristics, moods, etc.

This information is thus stored in the presence server based on publications received from the client or from the access network whenever any presence data for that client is introduced, updated, changed or deleted. According to common terminology in the field of presence services, a client that subscribes or requests for presence data is called the "Watcher", and a client that publishes presence data is called the "Presentity".

The "Presence Event Package for SIP" and the "Common Profile for Presence (CPP)" are extensions to SIP, enabling clients to publish and receive presence information as described above. Further, the SIP Presence Event Package has been adopted by the "Open Mobile Alliance (OMA)" and the 3GPP for use in IMS systems.

The SIP messages conventionally used in the presence context include "SIP publish" when the presentity sends presence data to the presence server for publication, "SIP subscribe" when the Watcher subscribes for presence data, and "SIP notify" when the presence server sends presence data to subscribing Watchers.

The present solution thus utilises the presence framework and the presence messages above can thus be used in a novel manner for conveying discovery information from one local network to another. Preferably, the service presence information may then be formatted according to the

regular Presence Information Data Format (PIDF) normally used for communicating presence data.

In order to convey the discovery information over the IMS network between the two local networks during the discovery process, each local network comprises a gateway node adapted to communicate the discovery information with the opposite gateway node over the IMS network using the presence framework. This gateway node will be called the "Service Discovery Gateway, SDG" in the following description. For example, the SIP presence event package mentioned above can be used as the framework for conveying the discovery information between the service discovery gateways.

One or both of the service discovery gateways are preferably further adapted to translate discovery messages between whatever local service discovery protocols are used within the local networks, and a generic service discovery format to be embedded in the presence framework. In this description, the term "generic format" indicates that the format is understood and supported by both service discovery gateways.

The service discovery gateway in each local network is further adapted to communicate discovery information in the generic service discovery format over the IMS network, where the discovery information (e.g. service descriptions and device capabilities) is embedded in regular messages of the presence framework. Hence, each service discovery gateway effectively acts as a gateway between whatever service discovery protocol(s) used within the respective local network and the presence messages (e.g. according to SIP) for the communication through IMS.

For example, the regular presence message "SIP subscribe" can be used to convey a request from one local network to obtain discovery information about devices in the opposite local network. Further, the regular presence message "SIP notify" can be used to convey announced discovery information about one or more devices in one local network to the other opposite local network. In addition, if a presence server is used in the IMS network for collecting and distributing discovery information by means of the presence framework, the regular presence message "SIP publish" can be used to convey published discovery information to the presence server.

Effectively, in presence terms, the service discovery gateway at either local network will thus act as the presentity when providing discovery information to the opposite network, and as the watcher when requesting and obtaining discovery information from the opposite network.

One or more service discovery protocols are used in each local network that are dependent on the type of devices and services in the local network. Hence, plural different service discovery protocols may be used for different devices within the same local network, where the service discovery gateway can translate each of them into the generic format and vice versa. The two service discovery gateways in the local networks can basically be configured in the same way logically, i.e. to both provide and obtain discovery information remotely, but may be implemented practically in different ways.

For example, one service discovery gateway may be implemented in a HIGA or RGW in a residential local network (e.g. using a service discovery protocol based on UPnP, Zigbee or IrDA), while the other service discovery gateway

in the opposite network could be implemented in a mobile user terminal temporarily being present in a local ad hoc network (e.g. using a Bluetooth-based service discovery protocol). In that case, the mobile user terminal should include an IMS client and functionality for translation between service discovery protocols, in order to act as a service discovery gateway and to communicate with the IMS network.

The mobile user terminal may thus have the functionality of a HIGA in order to set up IMS sessions on behalf of non-IMS devices in the ad hoc network, which is referred to as "PIGA Portable IMS Gateway". By implementing a PIGA and a service discovery gateway on a mobile terminal, it will also be possible to discover services and devices in such ad hoc networks, and provide information to remote service discovery gateways, or to an IMS-centric presence server, to publish and make such services available to other local networks.

The watching service discovery gateway in one local network may use the obtained discovery information to establish a service usage session with a device in the opposite local network. Both nodes must support the generic service description format to utilise the service information. The Service Usage protocol for a media session between two devices in opposite local networks (e.g. HTTP streaming, RTSP streaming, FTP, etc.) depends on the particular service and/or application, and it must be supported by both devices.

In Fig. 2, an exemplary scenario and process is illustrated for conveying discovery information between devices in a first local network 200 and a device in an opposite second local network 202 by means of a presence

framework over an IMS network 204, using a service discovery gateway at either local network. In this example, a media player 202a in the second network 202 will fetch media stored at a media server 200a in the first network, in order to present the media at the second network.

The local networks 200, 202 can be considered as limited service "islands" where different services available on individual local devices can be shared between the local devices within each service island. In this description, the term local network implies such a service island.

Thus, the first network 200 comprises a first service discovery gateway 200b and possibly an RGW 200c for communication of media outside the network 200, whereas the second network 202 comprises a second service discovery gateway 202b and possibly an RGW 202c as well. Each network 200, 202 must also have access to an IMS client, e.g. in a HIGA or an IMS user terminal, for communication over the IMS network 204, although it is assumed here that the IMS client resides logically in the shown service discovery gateways 200b, 202b.

As indicated in the figure, the first local network 200 uses one or more internal service discovery protocols SDP1, SDP2 valid within the network 200 for conducting discovery procedures. On the other hand, the second local network 202 uses another internal service discovery protocol SDP3 valid within the network 202 that may well be different and incompatible to the ones used in the first network 200, but not necessarily so. A discovery procedure can be conducted across the two local networks 200, 202 as described below.

The service discovery gateway 202b in the second network 202, using the address sdg2@yyy.com, may issue a

request for discovery information from the opposite network 200 embedded in the presence framework. The discovery information request can be sent in the form of a SIP subscribe message, effectively subscribing to "service presence events" from the service discovery gateway 200b in network 200. The SIP subscribe message could then be configured as:

Message example 1

```
SUBSCRIBE sip:sdg1@xxx.com SIP/2.0
From: <sip:sdg2@yyy.com>
To: <sip:sdg1@xxx.com>
Event: presence
Content-Length: 0
```

It is not necessary to further include a body in this message, although optional filters for the requested service presence information may be specified in the message. The SIP subscribe message above is thus directed to the service discovery gateway 200b in the opposite network 200, using the address sdg1@xxx.com, and the discovery process is therefore conducted "peer-to-peer", i.e. more or less directly between the two service discovery gateways 200b, 202b.

As mentioned above, it also possible to involve an intermediate presence server in the IMS network to collect and distribute service presence information between various local networks. Thus, using a presence server in the IMS network will basically enable discovery procedures across more than just two local networks. In this case, the presence server 204a has the address sps@xyz.com, and a SIP

subscribe message thereto from service discovery gateway 202b could then be configured as:

Message example 2

```
SUBSCRIBE sip:sps@xyz.com SIP/2.0
From: <sip:sdg2@yyy.com>
To: <sip:sps@xyz.com>
Event: presence
Content-Length: 0
```

The SIP subscribe message above is thus directed to the presence server 204a, and the discovery process is therefore considered to be "IMS-centric". As in the peer-to-peer case, it is not necessary to further include a body in this message.

Further, if the IMS-centric solution is used, the service discovery gateway 200b in the first network 200 can publish discovery information on devices in network 200 by sending a SIP publish message to presence server 204a, after having obtained the discovery information in a discovery process locally within network 200. First, the service discovery gateway 200b translates the locally obtained discovery information from a local service description format, if necessary, into a generic service description format understood by both service discovery gateways 200b, 202b. However, in some cases, the local discovery information can be embedded in the presence message without translation if already in a service description format understood by both networks, i.e. generic format, and the translation is not necessary. The SIP subscribe message from service discovery gateway 200b in the IMS centric case could then be configured as:

Message example 3

```
PUBLISH sip:sps@xyz.com SIP/2.0
From: <sip:sdg1@yyy.com>
To: <sip:sps@xyz.com>
Event: presence
Content-Length: entity-body-length
Content type: application/pidf+xml
<xml version="1.0">
```

- followed by a body containing the published discovery information according to the generic service discovery format, in this example in the XML (Extensible Mark-up Language) format. The discovery information is thus handled as service presence information according to the presence framework, in particular the Presence Event Package for SIP. The service presence information may then be formatted in line with the presence information data format (PIDF).

The SIP publish message above is thus directed to the presence server 204a in the IMS network 204 which then will distribute the published discovery information further by sending a SIP notify message to the subscribing service discovery gateway 202b in the second network 202. The SIP notify message from presence server 204a in the IMS centric case could then be configured as:

Message example 4

```
NOTIFY sip:sdg2@xxx.com SIP/2.0
From: <sip:sps@xyz.com>
To: <sip:sdg2@xxx.com>
Event: presence
```

Content-Length: entity-body-length
Content type: application/pidf+xml
<xml version="1.0">

- followed by a body containing the discovery information in the XML format as generic discovery information.

On the other hand, if the peer-to-peer solution is used, i.e. not involving the presence server 204a, the service discovery gateway 200b in the first network 200 can send a SIP notify message with discovery information directly to the subscribing service discovery gateway 202b in response to receiving the SIP subscribe message of example 1 above, and after having obtained the discovery information locally within network 200. The SIP notify message from service discovery gateway 200b in the peer-to-peer case could then be configured as:

Message example 5

NOTIFY sip:sdg2@xxx.com SIP/2.0
From: <sip:sdg1@yyy.com>
To: <sip:sdg2@xxx.com>
Event: presence
Content-Length: entity-body-length
Content type: application/pidf+xml
<xml version="1.0">

- likewise followed by a body containing the discovery information in the XML format as generic service discovery information.

In either case, the service discovery gateway 202b has now finally received the discovery information regarding devices in the first network 200, either directly from the

opposite service discovery gateway 200b or from the presence server 204a. That information can then be provided to any device in the second network 202 in a local discovery process, e.g. after translating the discovery information received in the generic format into some valid local service discovery protocol, if necessary. Further, if the status of a service is changed in network 200, service discovery gateway 200b (the presentity) can notify service discovery gateway 202b (the watcher) about the change as a "service presence event", e.g. depending on the events service discovery gateway 202b has subscribed to.

A procedure for enabling remote discovery of services and devices in a first local network from a location within a second local network, will now be described with reference to the flow chart in Fig. 3. The shown steps are executed by the service discovery gateway in the second local network.

In a first **step 300**, a request for discovery information on devices and services in the first local network is issued from the service discovery gateway in the second local network, over an IMS network using the presence framework. The discovery information request may be sent as a SIP subscribe message as described above for Fig. 2, either directly to a corresponding service discovery gateway in the opposite first local network or to an intermediate presence server in the IMS network, in order to subscribe to service presence information and events according to the presence framework.

In a next **step 302**, discovery information is received in a generic format over the IMS network, in response to the discovery information request. As explained above for Fig. 2, the discovery information may be received

either directly from the service discovery gateway in the first local network in the peer-to-peer case, or from a presence server in the IMS network in the IMS-centric case. The discovery information request may be received as a SIP notify message as described above for Fig. 2.

In a further **step 304**, it is determined whether it is necessary to translate the received discovery information from the generic format into a local service description format according to a local service discovery protocol used in the second network for service discovery procedures. If necessary, the discovery information is first translated in a following **step 306** into a local service description format valid for conducting local discovery procedures with at least one device in the second network. The translated discovery information is then finally provided to devices in the first local network during a regular local discovery procedure in a **step 308**. If it is not necessary in step 304 to translate the discovery information received in the generic format, the process can move directly to step 308 of providing the discovery information to devices in the first local network.

An exemplary messaging flow for remote discovery of services and devices in a first local network from a location within an opposite second local network, based on the above-described peer-to-peer method, will now be described with reference to the signalling diagram in Fig. 4. It should be noted that the skilled person will understand that each shown signalling step in the figure may represent one or more specific messages transferred back and forth according to the used protocol(s).

In a first **step 4:1**, a discovery process is at some point conducted within the first local network 400

involving a plurality of devices and services 400a and a service discovery gateway 400b, such that service discovery gateway 400b obtains or collects discovery information on the devices and services 400a. A local service discovery protocol is used in the discovery process and the discovery information obtained by service discovery gateway 400b may not be understood at the opposite network 402.

The second local network 402 comprises at least one device 402a and another service discovery gateway 402b. In a next **step 4:2**, service discovery gateway 402b sends a SIP subscribe message according to the presence framework directly to service discovery gateway 400b, as a request for discovery information on devices and services in the first local network 400. If necessary, service discovery gateway 400b translates, in a **step 4:3**, the discovery information obtained in step 4:1 is translated into a generic format that is understood by both service discovery gateways 400b, 402b.

Service discovery gateway 400b then provides the sends a SIP notify message in a following **step 4:4**, containing the translated discovery information, in response to the SIP subscribe message.

Then, at some later point sooner or later, another discovery process may take place within the first local network 400 in a further shown **step 4:5**, e.g. according to some predetermined scheme or whenever any device or service therein has been added, removed or modified, to update the discovery information. Since service discovery gateway 402b subscribes to service discovery information of network 400, service discovery gateway 400b may translate the newly obtained discovery information, in a further **step 4:6**, and

send the translated and updated discovery information to service discovery gateway 402b, in another **step 4:7.**

After either of steps 4:4 and 4:7, service discovery gateway 402b is able to conduct a discovery process locally within network 402 in order to provide the received discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network 402. Thus, service discovery gateway 402b may need to translate the received discovery information, as shown in **step 4:8**, from the generic format into a local service description format valid in network 402, before conducting a discovery process involving at least one device 402a in network 402, in a final illustrated **step 4:9.**

The example of Fig. 4 is thus based on the above-described peer-to-peer method where the management of "service presence events" is handled between the service discovery gateways 400b, 402b, although not involving any IMS network functions such as a presence server. In presence terms, the service discovery gateway 400b thus acts as a "service presence event handler". However, the SIP signalling between the service discovery gateways 400b, 402b will naturally pass over a suitable session control node in the IMS network, not shown. In particular, the IMS network can be used for identification, authentication, access control as well as addressing and mobility support of the end-to-end SIP messages.

Another exemplary procedure and messaging flow for remote discovery of services and devices, based on the above-described IMS centric method, will now be described with reference to the signalling diagram in Fig. 5. Again,

any signalling step shown in the figure may represent one or more specific messages depending on the used protocol(s).

Thus, two opposite service discovery gateways 500, 502 in a first and a second local network A and B, respectively, are both adapted to communicate with a presence server 504 in an IMS network, using SIP signalling according to the presence framework, to convey discovery information in a generic format understood by both service discovery gateways 500, 502. In this example, the shown process can be seen as having two parallel parts: one part marked "a" for making published discovery information from the first network A available in the presence server 504, and another part marked "b" for providing discovery information to the second network B.

The two-way arrows D in the figure generally represent a discovery process conducted locally within the local networks A and B, respectively. In a first **step 5:1a**, service discovery gateway 500 translates discovery information obtained in a discovery process D into the generic format, and sends a SIP publish message in a following **step 5:2a**, containing the translated discovery information, to presence server 504. The following steps of discovery D, translation **5.3a** and sending the SIP publish message **5:4a** are basically a repetition of the previous steps whenever the discovery information is updated. In this way, the presence server 504 will maintain discovery information on the first local network that is up-to-date.

The other part "b" of the process includes that service discovery gateway 502 sends a SIP subscribe message to presence server 504 in a **step 5:1b**, effectively requesting discovery information on devices and services in the first local network A. If service discovery gateway 500

has published any discovery information previously, the presence server 504 will send a SIP notify message in a **step 5:2b** containing the current discovery information in the generic format, e.g. according to the message example 4 given above.

In due time, the presence server 504 may send further SIP notify messages, as exemplified by **steps 5:3b** and **5:5b**, whenever the discovery information is updated by receiving SIP publish messages from service discovery gateway 500, as in steps 5:2a and 5:4a. In order to conduct a discovery process within the second local network B, the service discovery gateway 502 may eventually translate the obtained discovery information, as shown in a **step 5:4b**, into a local service description format according to a local service discovery protocol used in the second network B for service discovery procedures. A local discovery procedure D can then be conducted within the second network B, as indicated by the two-way arrow.

It should be noted that the process in part "a" of discovery D, translation (steps 5.1a, 5:3a) and sending the SIP publish message (steps 5:2a, 5:4a) can basically be executed regardless of the activities in part "b", i.e. only dependent on when the discovery information is updated. However, SIP notifying messages (e.g. steps 5:3b, 5:5b), may be sent from presence server 504 in response to receiving the SIP publish messages (e.g. steps 5:2a, 5:4a). Accordingly, after step 5:5b, a further translation step, not shown, may be executed in service discovery gateway 502 followed by another discovery process, and so forth.

The example of Fig. 5 is thus based on the above-described IMS centric method where the management of "service presence events" is handled by the presence server

504, effectively acting as a "service presence event handler" or "service presence event management server" in the IMS network.

One advantage of using an intermediate presence server for remote discovery, is generally that published discovery information of one presentity local network can easily be made available to any number of other watcher local networks, using the existing presence handling functionality. Different filters may also be applied for different watchers in the same manner as in regular presence procedures. Moreover, a presence server in an IMS network is generally deemed more powerful with respect to processing and storing capacity, as compared to resource-limited gateways and devices in typically small local networks.

Next, a more detailed description will be given of two service discovery gateways in opposite local networks, adapted to conduct remote discovery basically in the manner described above, with reference to the block diagram shown in Fig. 6. It should be noted that Fig. 6 illustrates different functional units purely logically, and the skilled person will be able to implement these functions in practice in any suitable manner, e.g. by means of hardware and software. As in the previous examples above, a first service discovery gateway 600 of a first local network, acting as a presentity, provides discovery information remotely to a second service discovery gateway 602 of a second local network, acting as a watcher, over an IMS network and using messages of the presence framework.

The first service discovery gateway 600 comprises a service and device discoverer 600a adapted to obtain discovery information by conducting a discovery process D within its local network, not shown. A translator 600b is

adapted to translate the discovery information obtained by the service and device discoverer 600a, if necessary, from a local service description format used in the first network into a generic format understood by both service discovery gateways 600, 602.

Service discovery gateway 600 further comprises a functional unit called the "presence presentity" 600c adapted to send the translated discovery information according to the presence framework, either in a SIP notify message directly to the opposite service discovery gateway 602 (the peer-to-peer case) or in a SIP publish message to an intermediate presence server 604 in the IMS network (the IMS centric case).

Further, the second service discovery gateway 602 comprises a functional unit called the "presence watcher" 602a adapted to receive the discovery information in the generic format in a SIP notify message, either directly from the presence presentity 600c of the opposite service discovery gateway 600 (the peer-to-peer case), or from the presence server 604 (the IMS centric case). A translator 602b is adapted to translate the discovery information received by the presence watcher 600a, if necessary, from the generic format into a local service description format used in the second local network.

The second service discovery gateway 602 further comprises a service and device announcer 602c adapted to announce the obtained discovery information to one or more local devices by conducting a discovery process D within the second local network, not shown. The transport of discovery information over the shown functional units, i.e. initially from the service and device discoverer 600a and finally to

the service and device announcer 602c, is generally illustrated by the arrows in Fig. 6.

The above-described functions in the two opposite service discovery gateways 600, 602 are illustrated here only for the case when discovery information from the first local network is presented to the second local network. However, each service discovery gateway 600, 602 may in practice comprise all the shown functional units for presenting discovery information the other way round, i.e. from the second local network to the first local network, to enable remote service discovery in both directions.

Fig. 7 and 8 illustrate examples of possible implementations of the functional flow in the above-described cases using the peer-to-peer and IMS centric methods, respectively. In Fig. 7, a first service discovery gateway 700 in a first local network thus provides discovery information directly to an opposite second service discovery gateway 702 in a second local network. In a first shown **step 7.1**, a presence watcher 702a in the second service discovery gateway 702 sends a SIP subscribe message to the opposite service discovery gateway 700, requesting for discovery information.

The SIP subscribe message is received by a presence agent 700c which may immediately respond by sending a SIP notify message back to presence watcher 702a, in a next **step 7.2**, containing any discovery information previously obtained in the second local network. The presence agent 700c is thus adapted to communicate presence messages of the presence framework with the presence watcher 702a.

In a further **step 7.3**, a discovery procedure is conducted within the first local network where a service

discoverer 700a in the first service discovery gateway 700 obtains discovery information from local service nodes 704. In this implementation, the service discoverer 700a is also adapted to translate the service description format used by each local service node into the generic service description format understood by the opposite service discovery gateway 702, although not shown here as a separate step.

The service information from the discovery process is then conveyed from service discoverer 700a to a presence user agent 700b in the service discovery gateway 700, in a **step 7.4.** The presence user agent 700b is adapted to prepare and manipulate presence information on behalf of a presentity, according to current standards. In this case, the presentity is actually the service discoverer 700a, and the presence information is the description (in SDP format) of a service. For example, the manipulation by presence user agent 700b may take care that this service information is transferred into a format that complies with PIDF.

Accordingly, presence user agent 700b conveys the prepared presence information to presence agent 700c, in a further **step 7.5.** Finally, triggered by this event of a new service presence information, presence agent 700c sends a SIP notify message to any subscribers (i.e. presence watchers) including presence watcher 702a, containing the discovery information in the generic format, over the IMS network using the presence framework, in a last **step 7.6.**

In the IMS centric case illustrated in Fig. 8, a first service discovery gateway 800 in a first local network provides discovery information to an opposite second service discovery gateway 802 in a second local network indirectly via a presence server 804. In this case, the first service discovery gateway 800 has omitted the presence agent 700c

and instead comprises an event publication agent 800a adapted to send published service presence information to the presence server 804. The first service discovery gateway 800 also comprises a service discoverer and a presence user agent just as in the previous example, although not shown here for simplicity. Thus, it is assumed that discovery information obtained in a local discovery procedure D is provided to the event publication agent 800a in the same manner as to the presence agent 700c in the example of Fig. 7, and description thereof will not be repeated here.

In a first shown **step 8.1**, a presence watcher 802a in the second service discovery gateway 802 sends a SIP subscribe message to a presence agent 804c in presence server 804, which may immediately send a SIP notify message back to presence watcher 802a, in a next **step 8.2**, as similar to steps 7.1 and 7.2 in Fig. 7. The presence agent 804c is thus adapted to communicate presence messages with the presence watcher 802a.

A further **step 8:3** illustrates that a discovery procedure is conducted within the first local network where a service discoverer, not shown, in the first service discovery gateway 800 obtains discovery information from local service nodes 806. After receiving the discovery information from a presence user agent, obtained and translated (if necessary) by the service discoverer, event publication agent 800a sends the discovery information as service presence information in a SIP publish message to presence server 804, in a following **step 8.4**. In this step, the published service presence information is received by an event state compositor 804a in the presence server, which is adapted to store received published presence information at a designated presence service. Thus, the received discovery

information is stored as updated presence information in a presence service unit 804b in presence server 804, in a next **step 8.5.**

The presence service unit 804b then informs the presence agent 804c that the presence status is changed, in a **step 8.6.** Finally, presence agent 804c sends a SIP notify message containing the updated discovery information to presence watcher 802a in the second service discovery gateway 802, in a last **step 8.7.**

The above-described functions in the service discovery gateway can be utilized in several ways, e.g. by specific applications that run on any type of devices in a local network. By way of example, two different use cases that can be implemented "on top of" the service discovery gateway will now be described, although other use cases are also possible within the scope of the present invention.

In a first example, the service discovery gateway SDG can be integrated with a service control application in a single user device in a local network, such as a mobile phone or an STB. For example, the SDG functions may be implemented in a local device also having a HIGA functionality, i.e. PIGA. An application (preferably having a dedicated logic and Graphical User Interface GUI) on the SDG device may utilise the above-described SDG functions, so that the user can actively control the discovery and usage of both local and remote services. Thereby, the user of the device can select an address of a remote local network (e.g. the user's home network while travelling) when located in a visited local network and discover services at the remote network (e.g. content in a media server). Moreover, the user can search for devices (e.g. a suitable media player) in a visited local network environment, and then initiate a

service usage session for a selected local device with a device in the remote network (e.g. a media streaming session between a remote media server and a local media player), using the above-described SDG functions.

In a second example, an SDG1 in a first local network acts as a presence watcher towards an SDG2 in a second local network. Thereby, SDG1 effectively acts as a "virtual" service provider of a service 2A from the second local network, transparently to the devices in the first local network. A control application with GUI could be provided e.g. by a service 1B in the first local network, and SDG1, utilizing the service presence information from SDG2, would appear as service 2A.

As another example, service 1A in the first local network (which could be a media provision service with control GUI) could search for available media clients in the first local network, and SDG1 could transparently appear as a virtual service 2A (which would be a media client) from the second local network. Hence, the remote media client (i.e. service 2A in network 2) could be utilized by the service 1A as if it were present within the same local network.

The above-described functions in the service discovery gateway SDG may also be utilised in the context of remote control and management of buildings. In the future, various equipment in private buildings can be connected to local home networks to support, e.g., the control of heating, ventilation, air conditioning, lightning and monitoring cameras. To control a building remotely, a remote control client, implemented in a PIGA, can connect to the SDG in the local network of that building (e.g. implemented within a HIGA) through the IMS communication infrastructure.

By using the present invention according to any of the above-described embodiments, the following advantages and improvements may be obtained:

By using the network infrastructure of IMS and presence framework as a generic transport means for remote discovery, local services can be provided and utilized across service islands in different local networks, even when various different device-specific service discovery protocols are used within the local networks. Thereby, a multitude of use cases and applications can be supported in a flexible and extensible way.

If an IMS centric presence server is used for the management of presence events in the remote discovery, the SDP traffic (in particular between the IMS network and the presentity SDG) can be optimised, and the end-to-end traffic between watcher SDGs and presentity SDGs can be minimized. Further, the generic format for service description may include metadata allowing the presence server to cache frequently used information etc.

Furthermore, when utilizing the presence framework in the manner described above, owners of services and content in a local network can define particular access rights to specific services and content in the local network, for users and user groups when located in other local networks. Also, the underlying IMS identification and authorization mechanisms can be used to control the services and content access.

When the "Presence Event Package for SIP" is used to exchange the discovery information through the IMS infrastructure by means of regular SIP messages, the inherent IMS/SIP addressing and mobility support can also be utilized.

While the invention has been described with reference to specific exemplary embodiments, the description is in general only intended to illustrate the inventive concept and should not be taken as limiting the scope of the invention. Although the concepts of IMS, HIGA, UPnP and SDP have been used when describing the above embodiments, any other similar suitable standards and network elements may basically be used for enabling the discovery process across local networks as described herein. The present invention is generally defined by the following independent claims.

CLAIMS

1. A method of enabling remote discovery of services and devices in a first local network from a location within a second local network, comprising the following steps executed by a service discovery gateway in the second local network:
- issuing a request for discovery information on the devices in the first local network, wherein the request is sent embedded in a presence subscribe message over an IMS network,
- receiving discovery information in a generic format embedded in a presence notify message over the IMS network in response to said request, wherein the received discovery information has been collected by a service discovery gateway in the first local network in a discovery process using a local service discovery protocol valid within the first local network, and
- announcing the received discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

2. A method according to claim 1, wherein the discovery information is translated from the generic format into a local format supported by the second local network, before being announced in the second local network.

3. A method according to claim 1 or 2, wherein the request for discovery information is sent to the service discovery gateway in the first local network, and the

discovery information is received as a response from that service discovery gateway.

4. A method according to claim 1 or 2, wherein the request for discovery information is sent to a presence server in the IMS network, and the discovery information is received as a response from that presence server.

5. A method according to claim 4, wherein the presence server has received said collected discovery information in a presence publish message from the service discovery gateway in the first local network.

6. A method according to any of claims 1-5, wherein the second local network is an ad hoc network and the service discovery gateway in the second local network is implemented in a user device acting as a multimedia gateway to access the IMS network on behalf of other devices in the ad hoc network.

7. A method according to claim 6, wherein a portable IMS gateway PIGA is implemented in said user device.

8. A service discovery gateway (602) for conducting remote discovery of services and devices in a first local network from a location within a second local network, the service discovery gateway being connected to the second local network, comprising:
- a presence watcher unit (602a) adapted to issue a request for discovery information on the devices in the first local network, wherein the request is sent embedded in a presence subscribe message over an IMS network, and

further adapted to receive discovery information embedded in a presence notify message over the IMS network in response to said request, wherein the received discovery information has been collected by a service discovery gateway in the first local network using a local service discovery protocol valid within the first local network, and

- an announcing unit (602c) adapted to announce the received discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

9. A service discovery gateway according to claim 8, further comprising a translator (602b) adapted to translate the discovery information from the generic format into a local format supported by the second local network, before being announced in the second local network.

10. A service discovery gateway according to claim 8 or 9, wherein the presence watcher unit is further adapted to send the request for discovery information to the service discovery gateway in the first local network, and to receive the discovery information as a response from that service discovery gateway.

11. A service discovery gateway according to claim 8 or 9, wherein the presence watcher unit is further adapted to send the request for discovery information to a presence server in the IMS network, and to receive the discovery information as a response from that presence server.

12. A service discovery gateway according to any of claims 8-11, wherein the second local network is an ad hoc network and the service discovery gateway in the second local network is implemented in a user device acting as a multimedia gateway to access the IMS network on behalf of other devices in the ad hoc network.

13. A service discovery gateway according to claim 12, wherein a portable IMS gateway PIGA is implemented in said user device.

14. A method of enabling remote discovery of services and devices in a first local network from a location within a second local network, comprising the following steps executed by a service discovery gateway in the first local network:
- collecting discovery information of said services and devices in the first local network in a discovery process using a local service discovery protocol valid within the first local network, and
- providing the collected discovery information in a generic format embedded in a presence message over an IMS network, thereby enabling a service discovery gateway in the second local network to announce said discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

15. A method according to claim 14, wherein the collected discovery information is obtained in a local format and translated into said generic format, before being provided over the IMS network.

16. A method according to claim 14 or 15, wherein the presence message is sent as a presence notify message to the service discovery gateway in the second local network.

17. A method according to claim 14 or 15, wherein the presence message is sent as a presence publish message to a presence server in the IMS network.

18. A service discovery gateway (600) for enabling remote discovery of services and devices in a first local network from a location within a second local network, the service discovery gateway being connected to the first local network, comprising:
- a discovery unit (600a) adapted to collect discovery information of said services and devices in the first local network in a discovery process using a local service discovery protocol valid within the first local network, and
- a presence presentity unit (600c) adapted to provide the collected discovery information in a generic format embedded in a presence message over an IMS network, thereby enabling a service discovery gateway in the second local network to announce said discovery information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

19. A service discovery gateway according to claim 18, further comprising a translator (600b) adapted to translate the discovery information obtained in a local

format into said generic format, before being provided over the IMS network.

20. A service discovery gateway according to claim 18 or 19, wherein the presence presentity unit (600c) is further adapted to send the presence message as a presence notify message to the service discovery gateway in the second local network.

21. A service discovery gateway according to claim 18 or 19, wherein the presence presentity unit (600c) is further adapted to send the presence message as a presence publish message to a presence server in the IMS network.

22. A presence server in an IMS network, for enabling remote discovery of services and devices in a first local network from a location within a second local network, comprising:
- an event state compositor (804a) adapted to receive discovery information on devices in the first local network, in a generic format embedded in a presence publish message from a service discovery gateway connected to the first local network, and
- a presence agent (804c) adapted to receive a request for said discovery information, embedded in a presence subscribe message, from a service discovery gateway connected to the second local network, and further adapted to send said discovery information embedded in a presence notify message to the service discovery gateway in the second local network in response to said request, thereby enabling the service discovery gateway in the second local network to announce said discovery

information to at least one device in the second local network, using a local service discovery protocol valid within the second local network.

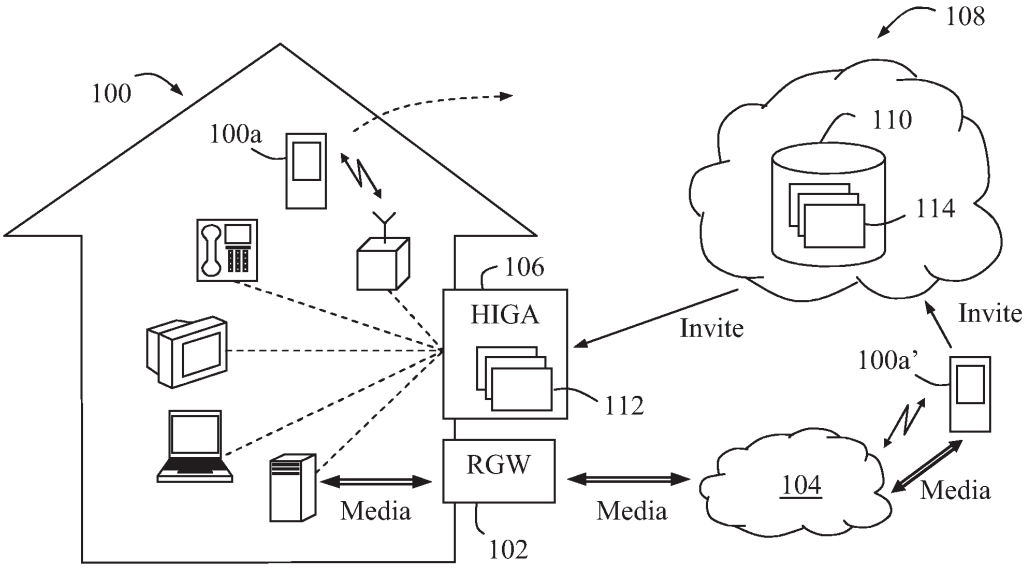


Fig. 1 (Prior Art)

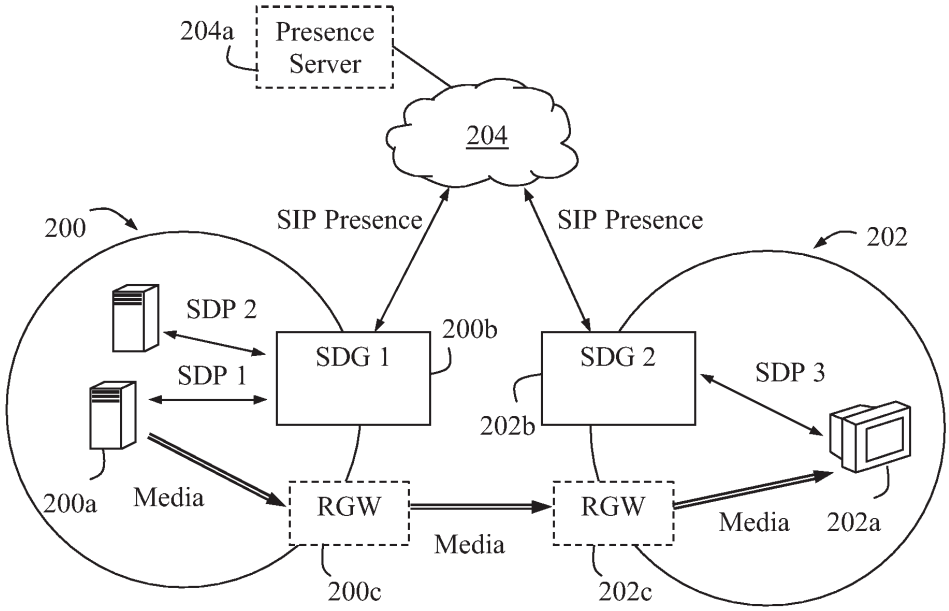


Fig. 2

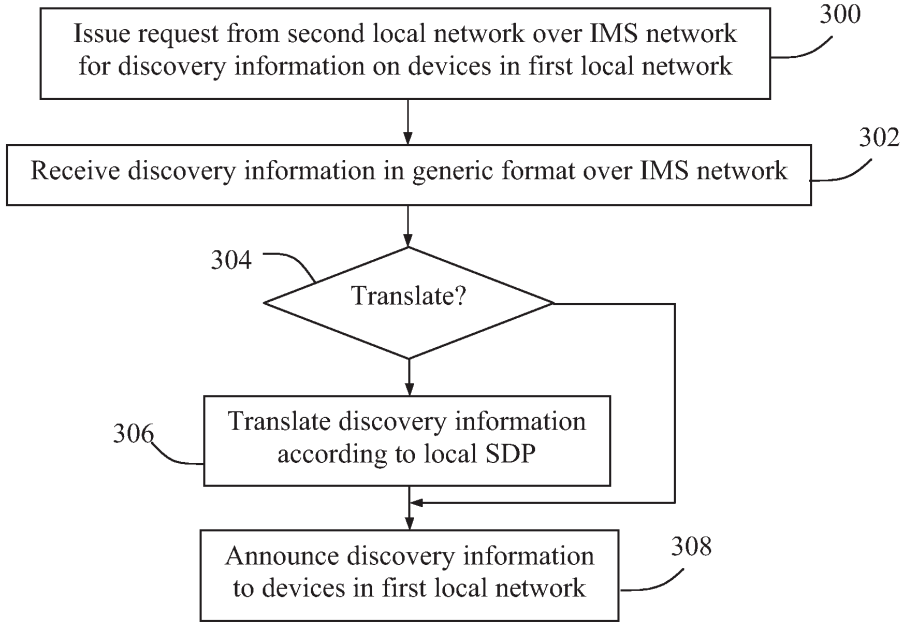

Issue request from second local network over IMS network for discovery information on devices in first local network
300
Receive discovery information in generic format over IMS network
302
304
Translate?
306
Translate discovery information according to local SDP
Announce discovery information to devices in first local network
308

Fig. 3

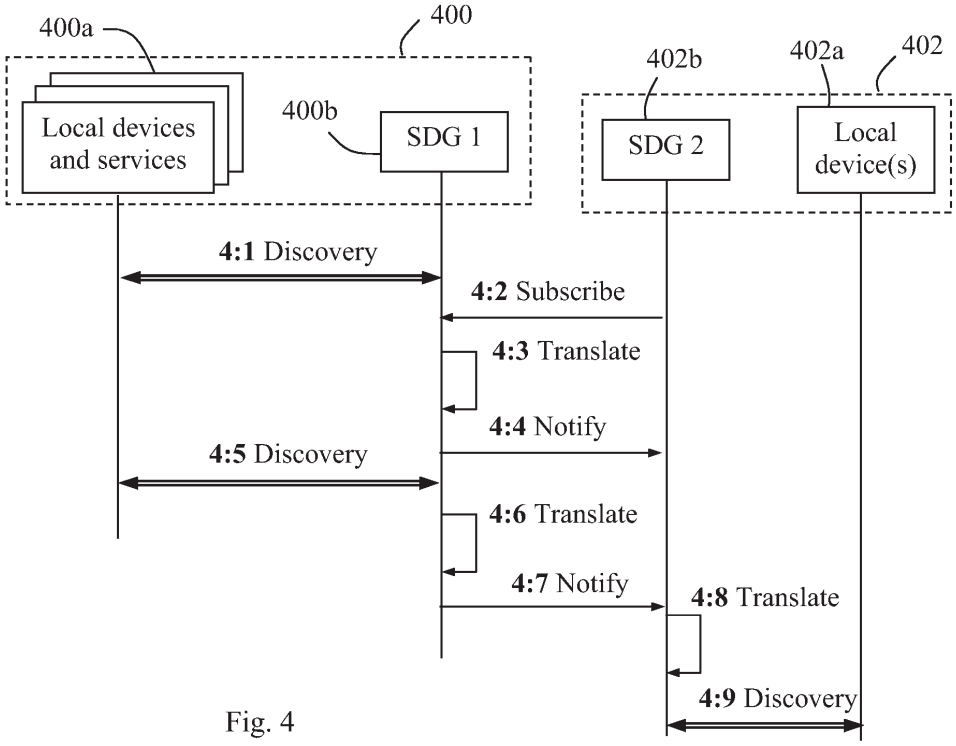

400a
400
Local devices and services
400b
SDG 1
402b
402a
402
SDG 2
Local device(s)
4:1 Discovery
4:2 Subscribe
4:3 Translate
4:4 Notify
4:5 Discovery
4:6 Translate
4:7 Notify
4:8 Translate
4:9 Discovery

Fig. 4

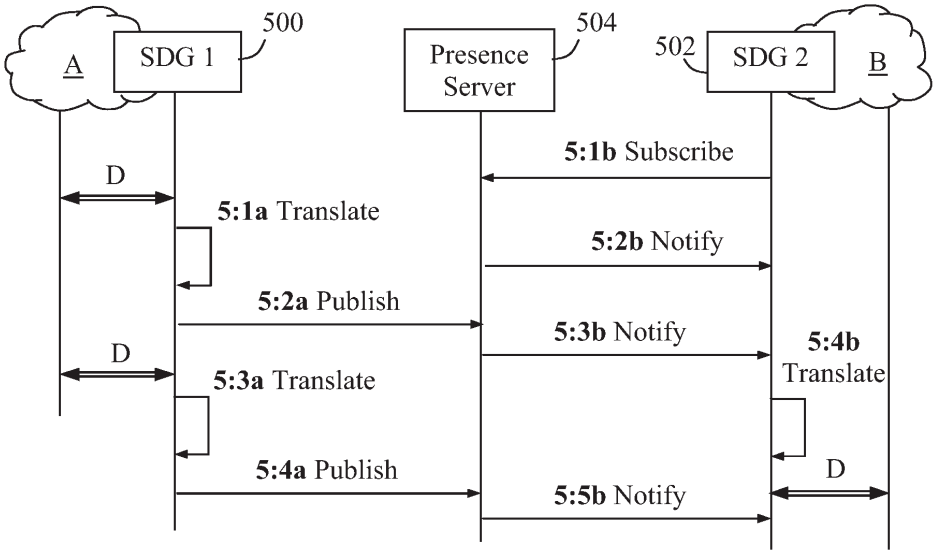

A
SDG 1
500
Presence Server
504
502
SDG 2
B
5:1b Subscribe
D
5:1a Translate
5:2b Notify
5:2a Publish
5:3b Notify
5:4b Translate
D
5:3a Translate
5:4a Publish
D
5:5b Notify

Fig. 5

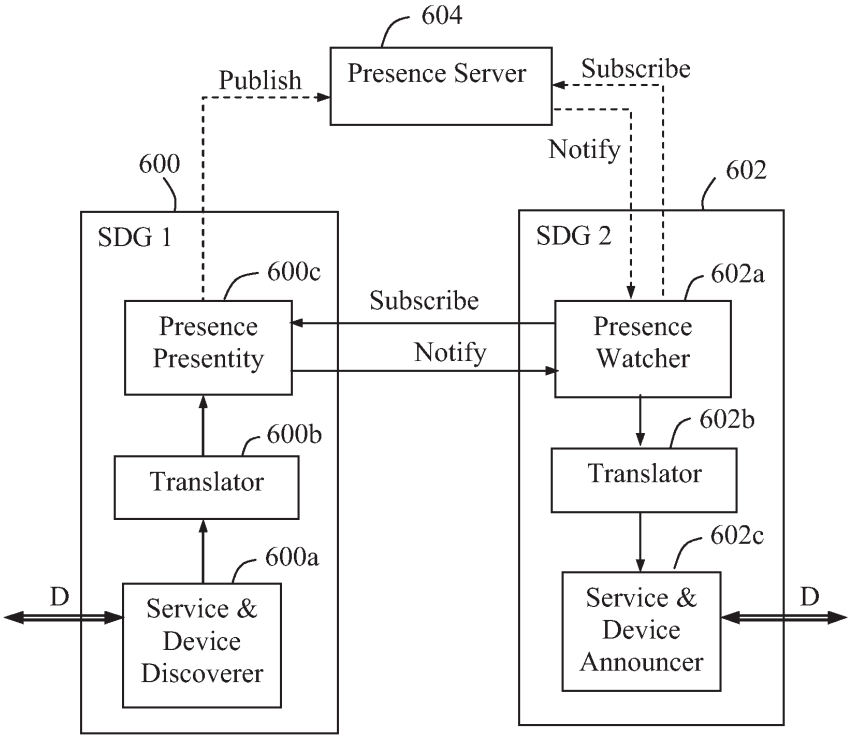

604
Presence Server
Publish
Subscribe
Notify
600
602
SDG 1
SDG 2
600c
602a
Subscribe
Presence Presentity
Notify
Presence Watcher
602b
600b
Translator
Translator
602c
600a
D
Service & Device Discoverer
Service & Device Announcer
D

Fig. 6

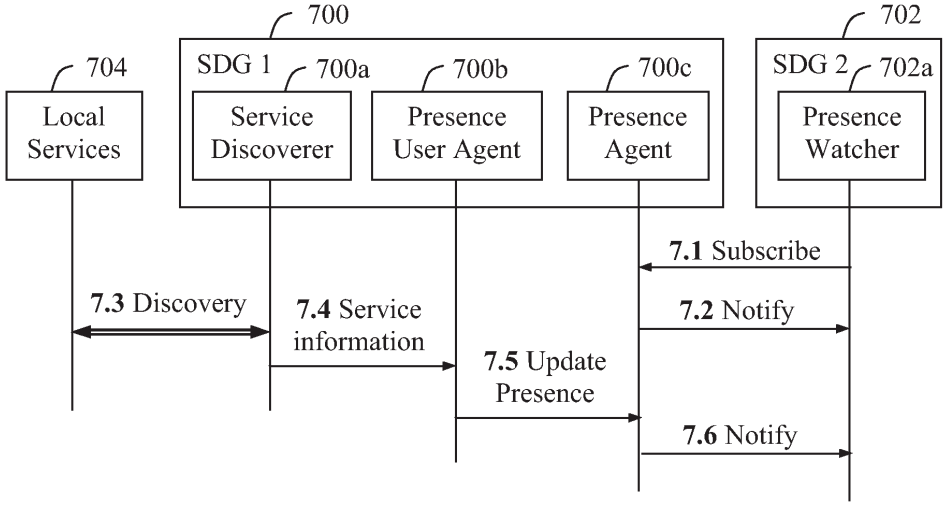

700
702
704
SDG 1
700a
700b
700c
SDG 2
702a
Local Services
Service Discoverer
Presence User Agent
Presence Agent
Presence Watcher
7.1 Subscribe
7.3 Discovery
7.4 Service information
7.2 Notify
7.5 Update Presence
7.6 Notify

Fig. 7

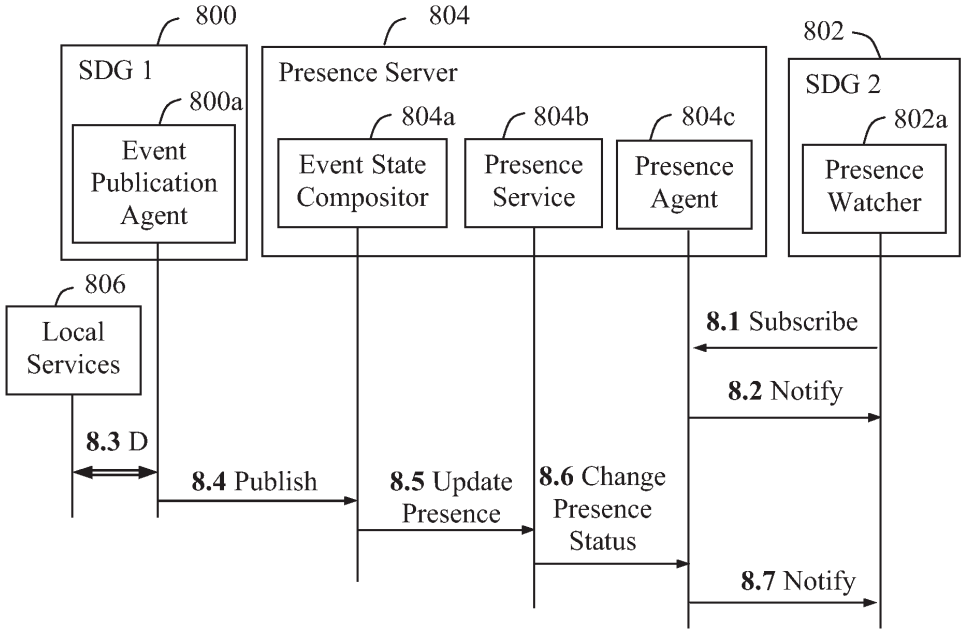

800
804
802
SDG 1
800a
Presence Server
804a
804b
804c
SDG 2
802a
Event Publication Agent
Event State Compositor
Presence Service
Presence Agent
Presence Watcher
806
Local Services
8.1 Subscribe
8.2 Notify
8.3 D
8.4 Publish
8.5 Update Presence
8.6 Change Presence Status
8.7 Notify

Fig. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE2007/050740

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 0215598 A1 (NOKIA CORPORATION), 21 February 2002 (21.02.2002), page 2, line 31 - page 5, line 4, figure 1, abstract --	1-22
A	WO 2006045706 A1 (TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)), 4 May 2006 (04.05.2006), whole document --	1-22
A	WO 2006079891 A1 (NOKIA, INC), 3 August 2006 (03.08.2006), whole document -- --------	1-22

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 May 2008	23-05-2008
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Oskar Pihlgren/CC Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (April 2007)

International patent classification (IPC)
H04L 29/06 (2006.01)
H04L 12/46 (2006.01)
H04L 12/66 (2006.01)

Download your patent documents at www.prv.se
The cited patent documents can be downloaded at www.prv.se by following the links:

- In English/Searches and advisory services/Cited documents (service in English) or
- e-tjänster/anförda dokument (service in Swedish).

Use the application number as username.
The password is **ONDOTNQCCQ**.

Paper copies can be ordered at a cost of 50 SEK per copy from PRV InterPat (telephone number 08-782 28 85).

Cited literature, if any, will be enclosed in paper form.

WO	0215598	A1	21/02/2002	AU	6839200	A	25/02/2002
				EP	1312225	A	21/05/2003
				JP	3805743	B	09/08/2006
				JP	2004507181	T	04/03/2004
WO	2006045706	A1	04/05/2006	CA	2583633	A	04/05/2006
				CN	101091374	A	19/12/2007
				EP	1805970	A	11/07/2007
				GB	0423845	D	00/00/0000
				GB	2419774	A	03/05/2006
WO	2006079891	A1	03/08/2006	AU	2006208939	A	03/08/2006
				EP	1844580	A	17/10/2007
				KR	20070091237	A	07/09/2007
				US	20060168656	A	27/07/2006