Network and Service Monitoring in Heterogeneous Home Networks

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 15 februari 2012 om 16:00 uur

door

Archi Delphinanto

geboren te Malang, Indonesië

Dit proefschrift is goedgekeurd door de promotoren:

prof.ir. A.M.J. Koonen
en
prof.dr. A. Liotta

Copromotor:

dr.ir. F.T.H. den Hartog

Cover design by S. Fitrianie
Author email: a.delphinanto@gmail.com

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Archi Delphinanto

Network and Service Monitoring in Heterogeneous Home Networks /
by Archi Delphinanto, Technische Universiteit Eindhoven, 2012.
A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-3094-6
NUR 959

Samenstelling promotiecommissie:

prof.dr.ir. A.C.P.M. Backx	Voorzitter
prof.ir. A.M.J. Koonen	Technische Universiteit Eindhoven, eerste promotor
prof.dr. A. Liotta	Technische Universiteit Eindhoven, tweede promotor
dr.ir. F.T.H. den Hartog	TNO, copromotor
prof.dr. E.W. Biersack	Institute EURECOM Sophia Antipolis
prof.dr. D. Marples MEng	Stirling University
prof.dr. J.J. Lukkien	Technische Universiteit Eindhoven
prof.dr.ir. E.R. Fledderus	Technische Universiteit Eindhoven

The work described in this thesis was performed in the Faculty of Electrical Engineering of the Eindhoven University of Technology and in TNO, and was financially supported by the Dutch Ministry of Economic Affairs through the Freeband B@Home project, and by the European Commission in the 7^{th} Framework Program through the Future Internet Gateway based Architecture of Residential Network (FIGARO) project.

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ - *in the name of Allah, the beneficent the merciful*

To ibu, bapak, Siska and Lavitanea

Summary

Home networks are becoming dynamic and technologically heterogeneous. They consist of an increasing number of devices which offer several functionalities and can be used for many different services. In the home, these devices are interconnected using a mixture of networking technologies (for example, CAT-5, wireless, coaxial cable, or power-line). However, interconnecting these devices is often not easy. The increasing heterogeneity has led to significant device- and service-management complexity. In addition, home networks provide a critical "last meters" access to the public telecom and Internet infrastructure and have a dramatic impact on to the end-to-end reliability and performance of services from these networks. This challenges service providers not only to maintain a satisfactory quality of service level in such heterogeneous home networks, but also to remotely monitor and troubleshoot them. The present thesis work contributes research and several solutions in the field of network and service monitoring in home networks, mainly in three areas: (1) providing automatic device- and service-discovery and configuration, (2) remote management, and (3) providing quality of service (QoS).

With regard to the first area, current service discovery technology is designed to relieve the increasing human role in network and service administration. However, the relevant Service Discovery Protocols (SDPs) are lacking crucial features namely: (1) they are not platform- and network-independent, and (2) they hardly provide information on the actual availability of resources in the network or a mechanism for (device) resource reservation. Consequently, devices implementing different SDPs cannot communicate with each other and share their functionalities and resources in a managed way, especially when they use different network technologies. As a solution to the first problem, we propose a new proxy server architecture that enables IP-based devices and services to be discovered on non-IP based network and vice versa. We implemented the proxy architecture using UPnP respectively Bluetooth SDP as IP- and non-IP-based SDPs. The proxy allows Bluetooth devices and UPnP control points to discover, access, and utilize services located on the other network. Validation experiments with the proxy prototype showed that seamless inter-working can be achieved

keeping all proxy functionalities on a single device, thus not requiring modification of currently existing UPnP and Bluetooth end devices. Although the proxy itself taxes the end-to-end performance of the service, it is shown to be still acceptable for an end user. For mitigating resource conflicts in SDPs, we propose a generic resource reservation scheme with properties derived from common SDP operation. Performance studies with a prototype showed that this reservation scheme significantly improves the scalability and sustainability of service access in SDPs, at a minor computational cost.

With regard to the second area, it is known that the end-to-end quality of Internet services depends crucially on the performance of the home network. Consequently, service providers require the ability to monitor and configure devices in the home network, behind the home gateway (HG). However, they can only put limited requirements to these off-the-shelf devices, as the consumer electronics market is largely outside their span of control. Therefore they have to make intelligent use of the given device control and management protocols. In this work, we propose an architecture for remote discovery and management of devices in a highly heterogeneous home network. A proof-of-concept is developed for the remote management of UPnP devices in the home with a TR-069/UPnP proxy on the HG. Although this architecture is protocol specific, it can be easily adapted to other web-services based protocols. Service providers are also asking for diagnostic tools with which they can remotely troubleshoot the home networks. One of these tools should be able to gather information about the topology of the home network. Although topology discovery protocols already exist, nothing is known yet about their performance. In this work we propose a set of key performance indicators for home-network topology discovery architectures, and how they should be measured. We applied them to the Link-Layer Topology Discovery (LLTD) protocol and the Link-Layer Discovery Protocol (LLDP). Our performance measurement results show that these protocols do not fulfill all the requirements as formulated by the service providers.

With regard to the third area, current QoS solutions are mostly based on traffic classification. Because they need to be supported by all devices in the network, they are relatively expensive for home networks. Furthermore, they are not interoperable between different networking technologies. Alternative QoS provision techniques have been proposed in the literature. These techniques require end-user services to pragmatically adapt their properties to the actual condition of the network. For this, the condition of the home network in terms of its available bandwidth, delay, jitter, etc., needs to be known in real time. Appropriate tools for determining the available home-network resources do not yet exist. In this work we propose a new method to probe the path capacity and available bandwidth between a server and a client in a home network. The main features of this method are: (a) it does not require adaptation of existing end devices, (b) it does not require pre-knowledge of the link-layer

network topology, and (c) it is accurate enough to make reliable QoS predictions for the most relevant home applications. To use these predictions for effective service- or content-adaptation or admission control, one should also know how the state of the home network is expected to change immediately after the current state has been probed. However, not much is known about the stochastic properties of traffic in home networks. Based on a relatively small set of traffic observations in several home networks in the Netherlands, we were able to build a preliminary model for home-network traffic dynamics.

Contents

Chapter 1

Introduction

A home area network or home network is a residential local area network or a collection of interconnected sub-networks. It is used for communication between devices deployed in the home. In this chapter, we introduce the research domain of home networking and frames the research work presented in the thesis. We discuss the trends, technologies, challenges and opportunities for enabling interoperability in home networks. Based on the current state-of-the-art, we then present our research questions and outline our approach toward solutions.

1.1 Home Network

Since its introduction, the Internet has organized the entire planet earth around telecommunicating networks of computers. Nowadays, most human activities depend on the timely across of information. In March 2011, the Internet World Stats (2011) reported that the Internet population (the amount of people using the Internet regularly) had surpassed 2 billion. The source also pointed out that this number is almost five times the population in the year 2000. As a result, the pace of innovation in Internet technology has accelerated. New technologies and applications are implemented to support the needs in education, work and leisure for more efficient and more exciting daily life.

Led by the rapid advancement of computer and Internet technologies, digitization of the home environment has blurred the boundaries between ICT media, broadcasting, and the field of audiovisual and consumer electronics. Many home appliances include the facility to connect to the home network as well as the Internet over which various services are provided. The home is evolving into a networked environment. A typical example is illustrated in Figure 1.1.

As data communication technology continues to penetrate the home and related prices are decreasing, a plethora of new intelligent devices are emerging, including, digital TVs, game systems, environmental controls, appliances, and safety and monitoring devices, along with lifestyle, wellness and medical devices. As a result, clusters of networked devices appear in the various domains of the user (home, car, office, work place, the personal operating space, etc). Consumers of various ages and backgrounds are becoming more receptive to connected products and experiences. Also telecom or access providers become interested in home networks as they would like to become "integrated providers", capable of reaching new areas in the home with advanced services and new networking solutions.

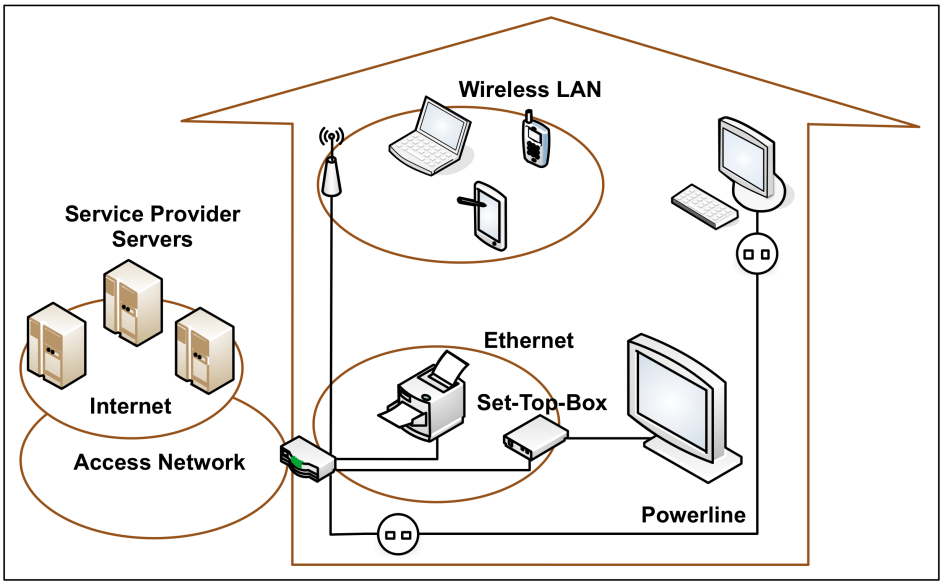


Figure 1.1: *Example of a home network configuration.*

This evolution has led to home networks that can often be characterized as dynamic and technologically heterogeneous. The growth of networked devices, in size as well as in number, leads to a significant management complexity. At the same time, the consumers become more demanding with regard to quality of experience and intuitiveness of the services offered by the provider. This challenges the service providers to maintain a satisfactory quality of service level in such heterogeneous network. This often leaves them puzzled with the huge variety of middleware specifications, initiatives, and consortia that exist in the home-network area.

1.1.1 Trend and Characteristics of Home Applications

Swanson & Gilder (2008) reported that the IP traffic of the U.S. alone in 2005 reached more than one Exabyte (10^{18} Byte) and predicted to reach an annual total traffic of one Zettabyte (10^{21} Byte) by 2015. The composition of applications that are predicted to load mostly the U.S. traffic in 2015 is shown in Figure 1.2. With U.S. traffic in 2011 around 13% (according the Internet World Stats (2011)), global IP traffic will probably pass the Zettabyte mark several years earlier.

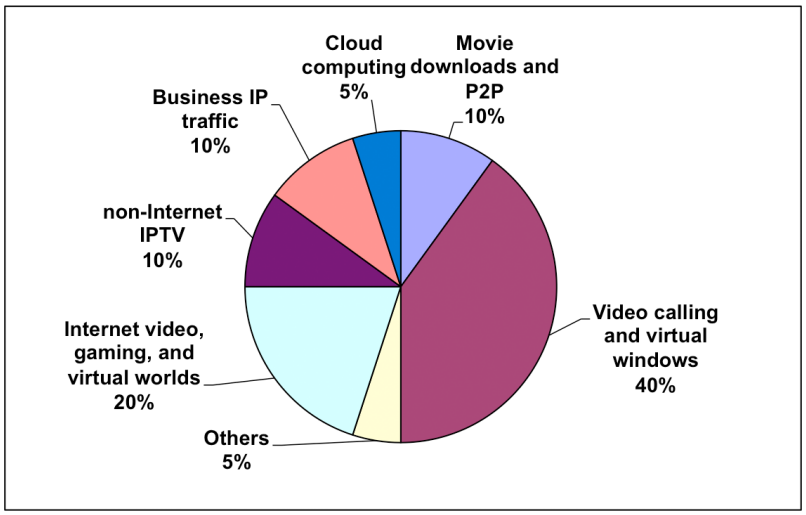


Figure 1.2: *The predicted applications that are to load mostly the U.S. IP traffic in 2015 (the data is imported and processed from Swanson & Gilder (2008)). "Others" include telephony, musics, photos and web.*

The prediction of Swanson & Gilder (2008) did not come without basis; a wide range of applications all of which use the home network at different levels have been identified. These applications come with many different names and can be categorized as follows.

1. *Connectivity* - This includes all applications aiming at the sharing of computing resources across multiple PCs and devices in the home, such as the sharing of data, files, peripherals (such as printers and scanners), mobile devices, home video recorders and digital cameras. Sharing a broadband Internet access connection and securing in-home communication are also considered part of this category.

2. *Entertainment and Information* - This includes interactive TV, streaming media, digital video recorders, and games. These applications are expected to be

the killer applications for the home network (supporting the prediction of Figure 1.2). Consumers are starting to demand easy access to their large collection of digital media files stored across multiple devices. This drives the need of network storage that can be accessed in real-time by various media players through different mediums (for example TV, PC, mobile phone, MP3 player, and the like).

3. *Communication* - This includes instant messaging, chat alerting, Voice over IP (VoIP), videoconferences, video calls and video game player communication. Such communication services can be offered in real-time and high quality and at lower cost, thanks to the vast transformation of the telecommunication industry in the past decade enabling telecommunication networks to carry multiple types of communication (data, voice and video) on the same circuits.

4. *Home Automation* - This includes applications that interconnect security, lighting, heating systems, home ventilation and air conditioning, and other home control systems for the purpose of user's convenience and energy management. These applications are often labeled as *smart-home* applications.

5. *Home Monitoring* - This includes applications that monitor one or more objects in the home from any external location through an Internet connection. Applications that fall in this category are demand-side energy management, health care monitoring, remote security monitoring, web-based surveillance cameras, and the like.

Until recently, applications such as home automation and home monitoring were available only to the high-end consumer market. However, they gradually move toward a larger segment of residential consumers (Duenas et al., 2005).

1.1.2 Quality of Service Requirements and Provision

Many home applications run on top of the Internet Protocol (IP), mainly to gain maximum interoperability over a diversity of underlying link-layer technologies. IP was designed to provide "best-effort" services for the delivery of data packets and to run across virtually any network transmission media and system platform. This means the network cannot give any guarantee regarding the quality of the transmission. Therefore, any application that runs over IP needs to configure the relevant network quality of service (QoS) for being "acceptable" to its end-users. The QoS parameters can be expressed using intrinsic metrics, such as:

- *Delay*, the time needed to send a packet,

- *Jitter*, the variation of delay,

- Required *bandwidth*, the minimal data rate for sending raw data and overhead,
- *Loss*, including *packet error-ratio* (i.e. the ratio of errored packets of all received packets) and *packet loss ratio* (i.e. the ratio of lost packets from all packets transmitted in population of interest).

The requirements for the QoS parameters differ, depending on the application type. Table 1.1 gives an example of QoS requirements for different application types. These requirements are considered critical for the application to run successfully. In the example, video broadcast application with high definition (HD) quality appears to have the strongest QoS requirements and consequently yields a great necessity of a QoS provision mechanism in the home network.

Furthermore, the QoS parameters are often used by service providers for defining and measuring Quality of Experience (QoE). QoE is a subjective measure of a customer's experiences with a service (web browsing, phone calls, TV broadcast, etc). QoE requirements for broadband triple-play applications are defined by Broadbandforum.org (2006c)

Table 1.1: *QoS metrics for different application types, derived from Ji et al. (2004).*

Application	**Bandwidth (Mbit/s)**	**Delay (millisecond)**	**Jitter (microsecond)**	**Packet Loss Ratio**
High Definition (HD) Video Streaming	13-24	100-300	0.5	10^{-9}
Standard Definition (SD) Video Streaming	3-6	100-300	0.5	10^{-9}
DVD Quality Video Streaming	6-8	100-300	0.5	10^{-9}
Voice over IP (VoIP)	< 0.064	10	-	10^{-2}
Network Gaming	< 0.1	10	-	10^{-9}
Internet Video Conferencing	0.1-2	75-100	-	10^{-3}

Many QoS solutions for IP networks are already available, but seem not to be suitable for home networks. Most of them operate on the principle of traffic classification (Delphinanto et al., 2011b), where each data packet is placed into a limited number of traffic classes, and each router on the network is configured to differentiate the traffic based on its class. Such solutions need to be supported by every device in the end-to-end path to be effective. This makes them relatively expensive for consumers with many non-depreciated devices: to enjoy QoS, they have to buy new devices. Besides, home networks are likely composed by different layer-2 technologies (such as wired LAN, WLAN, and power-line communications) that need different QoS solutions. One

of the reasons is that the traffic classifiers of the layer-2 technologies are defined differently. Therefore, intermediate translators will be needed to guarantee an end-to-end QoS in a heterogeneous path.

Alternatively, QoS provision can be based on measuring and predicting some intrinsic parameters of the network in real time, and reporting the information to a relevant application. The application can then pragmatically adapt its properties to the actual condition of the network, and report intermittent issues to the user. In other words, instead of trying to make the network application- and content- aware, it seems more feasible to make the applications network aware, real-time. This solution is often easier to apply to current home networks than the ones based on the traffic classification, because the solutions only require software updates, and not necessarily on all devices. A crucial part of these solutions are measurement and predicting tools for the real-time assessment of end-to-end available bandwidth (throughput) between the relevant client and server in the network. To obtain the information real-time, the tools will need to probe the network frequently.

1.1.3 Home-Network Technologies

There are many network technologies available for the home network. A decade ago, Zahariadis et al. (2002) had reported that it was more than 50 candidate technologies and standard specifications for home networking existed. These technologies can be grouped into three broad groups, namely as: (1) new wiring, (2) existing wiring, and (3) wireless. The last two groups are often categorized as no new-wiring. Their characteristics and examples are given in Table 1.2.

The technologies with new wiring require a new structured wiring system in the home. These technologies provide a safe way to deploy new services in the home environment as they can support high data-rate, high privacy and security (due to closed networking environment), and are endorsed by global standardization (i.e. IEEE). Examples of technologies that fall into this category include Ethernet, Universal Serial Bus (USB), and IEEE 1394. However, installing new wires in an existing home is not a solution amenable to the mass market. Most consumers are unwilling or cannot afford the large-scale rewiring of their home. As a result, the attention of home networking has been put on solutions that eliminate the need for installing wires in the home.

As the first solution, home networking is established using existing in-home wiring systems such as powerline, phoneline, or coax cable. Examples of technologies falling in this category are HomePNA v3.1 (over phoneline or coaxial cable), MoCA (over coaxial cable), and HomePlug AV (over powerline). With respect to the powerline technologies, due to their characteristics and relatively ease of installation, these technologies are often chosen by service providers to extend connectivity coverage in the

Table 1.2: *Home-networking characteristics based on physical media.*

	New Wiring	**Existing Wiring**	**Wireless**
Instalation cost	High	Low	Low
Best uses	New construction and remodeling	Interconnecting stationary devices	Mobile devices
Coverage area	Wherever needed	Multiple electrical outlets in each room, several rooms with telephone outlets; few rooms with coax outlets	Ideally throughout home
Quality of service support	Not inherent in the standards	Inherent in the standards	Not inherent in the standards
Privacy and security	High	High	Medium
Standard compliance	IEEE	ITU, Home PNA, Home Powerline	IEEE
Medium	CAT5	Phone line, Coax cable, Powerline	Radio (especially in 2.4 GHz)
Technology Example (maximum data-rate in Mbit/s)	IEEE 802.3 (100), IEEE 802.3ab (1000)	HomePNA 2 (320), MoCa 1.1 (175), HomePlug AV 2 (640)	Bluetooth 2 (2.1), IEEE 802.11b (11), IEEE 802.11g (54), IEEE 802.11n (72.2).

Legend: AV = Audio-Video, MOCA = Multimedia over CoAx, and HomePNA = Home Phone wire Networking Alliance.

home.

Another solution is with the use of wireless technologies such as IEEE 802.11x and Bluetooth. This solution gains quickly a wide acceptance of home consumers, mainly due to the largest location flexibility and no requirement to any wiring system. However, current wireless technologies for the home networks are inherently prone to link quality degradation due to interference in the unlicensed 2.4 GHz / 5 GHz bands they operate in.

Choosing the right home-networking technology is critically important to the long-term of end-to-end service delivery. The choice is rapidly transitioning from a consumer centric model to the domain of service providers. Despite best efforts to single-out a predominant technology with which to deploy services, many service providers choose to a hybrid network solution. A consequence of this fact is that many customer premise equipments, particularly residential gateways, have been manufactured with more than one advanced home-networking technology on board, in addition to stan-

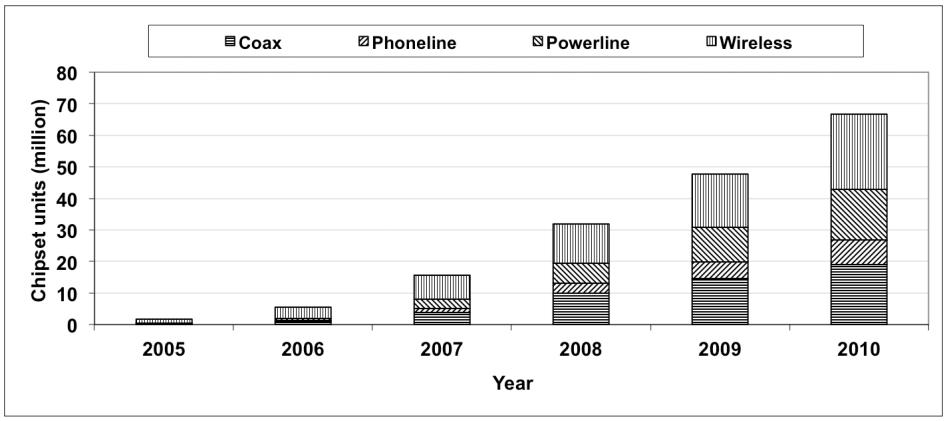


Figure 1.3: *Worldwide Total Available Market (TAM) of Home-Network Ports by interface type (Mocalliance.org, 2007).*

dard (new wiring) RJ-45 Ethernet ports.

Another consequence, the implementation of the no new-wiring technologies in the home is predicted to grow rapidly. Figure 1.3 shows the chipset unit (being manufactured and the share of the no new-wiring technologies in the global market from 2005-2010 (Mocalliance.org, 2007). The total chipset units of all technologies increased significantly (over 41 times) within five years, namely over 1.6 million units in 2005 to 6.6 million units in 2010. The figure also shows that the wireless solutions dominated the market and are also expected to dominate for the coming years. In addition, the rapid growth of the coax-based solutions was mainly influenced by the North America market where pre-installed coaxial wiring is already in place in well over 90% homes. The low growth of the phoneline solutions was because they have only limited traction in IPTV deployment in those periods, notably in eastern European and some in Asia.

1.1.4 Home-Network Architecture

Given the large heterogeneity of home-networking technologies and devices in a single home, and compared between different homes, it will be challenging to define a home-network architecture that can represent all device functionalities and connections. We follow Figaro Project (2011) for such architecture, which is shown in Figure 1.4. The architecture contains all generic building blocks needed to describe devices and their connections for home applications. This includes applications, which are completely contained within the home (such as a local multimedia server for home entertainment), as well as applications depending on external services (such as IPTV, remote patient monitoring, and the like). This architecture also accommodates all

home-network connectivity model postulated by the Home Gateway Initiative (HGI, 2006). The HGI is an industry consortium that defines technical requirements for home gateways.

In the architecture (Figure 1.4), a home network may be connected to multiple Service providers (SP) that provide services (for example, a Web service, ISP service, TV, voice, energy service and e-health service) typically from outside the home to end user devices (EUD) in the home. A home network may have several Internet access subscriptions. However, as an initial approach the reference architecture assumes only one Access Network (AN) provider enters the home. The End User Devices in the home may provide in-home services to other EUDs in the home (for example, home automation applications) or to the Internet. Besides EUDs, the reference architecture also defines Home-Network Infrastructure Devices (HNIDs), Peripheral Devices (PDs) and a Residential Gateway (RG) in the home. The terms home gateway (HG) and RG are used interchangeably. The term device is also used interchangeably with the term resource, which is any source of supply, can specifically consist of files, file-system, operating system, processing capability, communication capability, visualization of audio and/or video, etc.

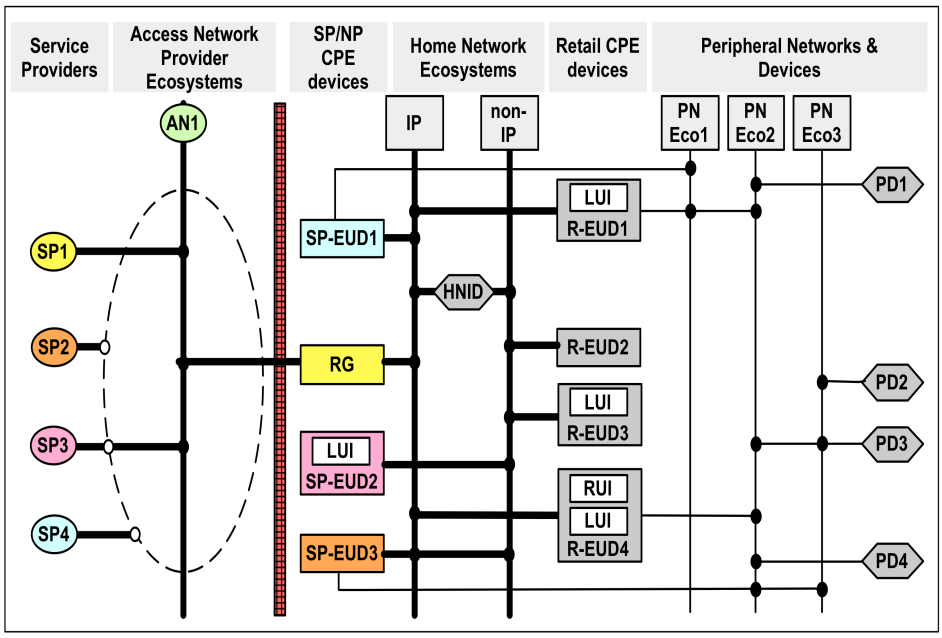


Figure 1.4: *The architecture of the home network based on Figaro Project (2011, p. 10).*

The following definitions are used to describe the infrastructure and networks in

Figure 1.4.

- *Access Network* (AN) physically connects EUDs or HNIDs to one or more SP's servers in the service delivery network outside the home, for example, ADSL, Cable, or FttH.
- *End-User Device* (EUD) is operated by the end user and offers a User Interface (UI) to the end user through which he can access services. The EUD can be managed and owned by:
  - Service provider, namely SP-EUD; for example, a Set-Top Box, smart meter or an RG. SP-EUD is usually leased or given away for free by service providers as part of the service agreement.
  - End-user or consumer, namely Retail EUD (R-EUD); for example, a TV, PC or electronic weighing scale. The consumer buys these devices from retail shops that are independent from the SP.
- *Home Network Infrastructure Device* (HNID) is an EUD that does not offer direct services to the end-user. However, the device is important for end-to-end service delivery. Examples are network terminators, bridges, switches, and routers. HNIDs can also provide the interconnection between multiple (home) networks. The RG typically contains HNID functionality.
- *Residential Gateway* (RG) is typically one device (but can be constructed of more devices) that connects one or more access networks to the home network and delivers services to the home environment.
- *Home Network* (HN) physically interconnects (indicated via thick line and black dots) EUDs and HNIDs inside the home. It can be interconnected using HNIDs as well. Here, IP and non-IP networks are distinguished.
- *Virtual Backbone* (VB) virtually connects (indicated via dotted oval, thick line and white dots) an AN to a global infrastructure like the Internet, connecting NPs and SPs.
- *Peripheral Network* (PN), such as USB, HDMI, and Bluetooth physically connects (indicated by thin line and black dots) PDs to a master device. PDs typically connect to one master and do not exist standalone. The master may aggregate PD device functionality and make it accessible over the HN. Some PN technologies can in principle also be used for non-IP HN control networks, for example Bluetooth, and IEEE 1394. EUDs and HNIDs can have both PN and HN interfaces.

- *Local User Interface* (LUI) is a user interface on the device itself (buttons, display, etc.). In the remainder of the document we only discuss Remote User Interfaces (RUIs), which are exported to other devices.

The colours in Figure 1.4 designate who has the ownership/control over a network or device. They relate to the business models and the stakeholders. Each colour indicates a different business role. A given business party may cover one or more business roles. For instance, it is assumed that SP1 (such as an ISP) owns, controls and manages the RG, but does not own, control, and manage the public access network (which in turn is owned, controlled, and managed by NP1). Grey indicates that the device or network is owned, controlled and managed by the end user. These devices and networks are typically bought in retail shops. The end user may decide to outsource the management and control of some of his devices to other stakeholders, for instance a service provider. It is also important to realize that in practice the devices RG, EUD, HNID, and the like may be combined in a single physical device in any combination. For instance, a Network Attached Storage (NAS) may also contain an Ethernet switch.

1.2 Home Gateway

During the last few years, the concept of Home Gateway (HG) has evolved. Earlier, the term was applied for just bringing IP connectivity to the home. Available services were common application-level programs such as Web or email. Recently, the rapidly growing business opportunities in the Internet have driven operators to introduce some value-added services such as Multicast TV and VoIP. Consequently, HG has become increasingly more intelligent and technically advanced. The HG now performs various tasks, ranging from protocol translation to QoS enforcement and enabling remote access. Since most of these tasks are dynamic, configurable, and manageable, and can change over time, they should be implemented by updateable software.

1.2.1 Standardization

Standardization of HG technology is a crucial development to enlarge the market of specific HG implementations while at the same time keeping costs down. The *Open Services Gateway Initiative* (OSGi) is a consortium founded in 1999 to create open specifications for the delivery of a wide array of services to end-users. The consortium has defined the OSGi service platform that is a container built on top of a Java virtual machine. The platform hosts the deployment of applications, services or components units called bundles. It supports multiple services and collaboration among services, provides security and simplifies the works of service administration by allowing users and service providers to administrate their bundles remotely and providing automatic

checking/resolving of the dependency of these bundles. OSGi was primarily developed for home gateways but is now also developed on many other devices such as mobile phones and car gateways. Key features offered by the OSGi platform are: (1) it is platform and application independent, (2) it supports multiple services and collaboration among services, (3) it provides security, and (4) it simplifies the works of service administration and improves stability.

In 2004, a telecom driven consortium founded a consortium so-called Home Gateway Initiative (HGI). The HGI works to facilitate the development of an HG device that matches the business requirements of telecoms operators; shortening the time to market of their services whilst providing an improved customer experience by seamlessly integrating a large variety of devices, networks and services into a virtually homogeneous, easy to use broadband home environment. The HGI has defined operating system requirements for HG, which are largely based on the OSGi specifications. They operate on three levels. The first one is software management, which includes:

- Configuration of software pieces called modules.
- Management of their life cycles: install, update, uninstall.
- Dynamics and security enforcement: modules should be registered and verified; their dependencies resolved, and then linked.
- Resets for default configuration parameters, and for firmware debugging in case of low-level failures.

The second level is performance management and diagnostics, which includes:

- Remote diagnostic tests for hardware and software elements.
- Performance monitoring.
- Support for events sent by the gateway.

The third and last level gathers definitions of users in presence:

- A super-user, in control of everything that is manageable.
- A local administrator, in control of local management (for example firewall and end-users).
- End users, with permissions set by the local administrator.

1.2.2 Remote Management

One of the most essential topics being dealt with in HGI is remote management of HG and other Customer Premises Equipment (CPE) in home networks behind the HG. The remote management is essential for service providers because of the following considerations. One of the considerations is service providers want to guarantee end-to-end service quality, not only for services delivered to officially "provider-supported" end devices, but also to devices that the customer bought off the shelf. Another consideration is the home network will remain as technically heterogeneous as it is to day, or worse. This does not just concern the various kinds of transmission technologies, but also a manifold of control and management protocols. If there is a problem in these networks, it will lead to end users calling the service providers' help desks. Although service providers posses tools to manage their own core and access networks, they lack the means to gather information related to home-network characteristics. For them, the home network is largely a black box.

From a telecommunications management point of view, the HG is nothing more or less than a network element in the telecommunications network. For network element management, one of the most popular management protocols of the last decade is the Simple Network Management Protocol (SNMP) (IETF.org, 2002). However, problems arise for which SNMP seems not to be cut out. There is not a widely accepted architecture for the delegation of management scripts; local Management Information Bases (MIBs) are vendor specific; security issues are still open; and network administrators do not rely on SNMP to perform configuration management of end devices (Neisse et al., 2004). Therefore, web-services based management has become increasingly popular.

One of these web-services based protocols is the CPE Wide Area Network (WAN) Management Protocol (CWMP). This has been developed for the sole purpose of HG management and is defined in TR-069 (Broadband-forum.org, 2007). Its primary capabilities are secure auto-configuration and dynamic service provisioning, software or firmware image management, status and performance monitoring, and diagnostics. TR-069 defines Remote Procedure Calls (RPCs) and, importantly, also standardizes the data model of the HG in TR-098 (Broadband-forum.org, 2006b). Security is considered by using Secure Socket Layer (SSL) or Transport Layer Security (TLS) technology. Another advantage of CWMP over SNMP is the ability of the CPE to initiate a TR-069 management session. In TR-069, the remote management server is called Auto Configuration Server (ACS). TR-069 is currently gaining wide acceptance with service providers. This is also the reason that our solution in this work is also based on TR-069. This is to avoid further diversity, and thus complexity, in the remote management solutions.

1.3 Service Discovery Technology

Service discovery technology is designed to configure a network dynamically, discover the device- and service- properties, of the devices present, and communicate these properties to other devices in the network. This technology includes protocols, data structures, and algorithms that enable service-users (*client*) to seek and retrieve particular services of interest (for example, for printing and displaying) and *devices* providing those services (for example, printer and TV) to advertise their capabilities to the network. Without manual configuration and device driver installation, the technology is developed to gradually relieve the human role in increasingly complex network and service administrations. This technology also allows the network to be self-healing by automatic detection of services that have become unavailable. Once services have been discovered, devices in the network could remotely control each other's services by adhering to some standard of communication. All of these functionalities are encapsulated in middleware and a so-called as service discovery protocol (SDP).

Currently, there are many SDPs available, such as Universal Plug and Play (UPnP), Bluetooth SDP, and Jini. With the given functionalities, SDPs can minimize the complex administration of already devices and services in the home network. Therefore, many home devices support one of the SDPs. Also resource lean devices can in principle take part in service discovery activities, namely by delegating some of their load to more powerful devices.

1.3.1 Requirements for the Home Network

Inspired by previous work of Teger & Waks (2002); Sundramoorthy et al. (2003); Duenas et al. (2005), and the characteristics of the home network as described in Section 1.1, we summarize the requirements that home network put to SDPs as follow:

1. The SDP should be able to run over heterogeneous networking technologies that are available or common in the home, such as Ethernet or wireless LAN. Here, support of security should be included. Many SDPs such as UPnP or Bluetooth SDP rely on the security mechanism of the link-layer or application layer.

2. The service discovery scope (namely the multitude of service and device types supported) and the scalability of the SDP (namely the amount of devices and services it can handle in a single network) should be suitable future home networks.

3. The SDP should support the provision of link quality of service (QoS), especially for applications such as video, etc. In addition, using the SDP itself should not lead to reduction of quality of user experience.

4. Using the SDP should not generate much traffic in the home network, and as such adding to the trend of the home-network traffic already growing rapidly.

5. Standardization of the SDP should include a data structure for the various services the SDP is expected to support. This is to guarantee compatibility of operations of devices manufactured by different companies.

1.3.2 Existing SDPs

The role of SDPs in computer networks is crucial. Many SDP surveys have been carried out to understand the capabilities of existing SDPs, such as Zhu et al. (2005); Meshkova et al. (2008). The features of existing SDPs are shown in Table 1.3 and matched to the requirements as given in Section 1.3.1. Their fulfillments to the requirements are indicated by light-grey cell. For requirement number 4, its fulfillment can be determined by the SDP architecture, which can be *distributed* or *centralized*. A distributed architecture allows service users (client) and services to find each other directly, namely using a multicast mechanism. In a centralized architecture, there is a third component, repository, which functions as a service catalogue. Due to the multicast mechanism, a distributed architecture leads to increased traffic and is therefore more intrusive than a centralized architecture. However, a centralized architecture introduces a single bottleneck and single point of failure.

1.3.3 Challenges

Table 1.3 shows that none of the SDPs can satisfy all requirements. Different SDPs have similar high-level goals, but they have quite different architectures. Devices using service discovery, will often use just one of these protocols or another analogous one. This means that clients and services that are using different technologies, will not be able to cooperate. Since it is likely that several protocols will be widely used, there will be a need for an interoperability solution that allows clients and services using different service discovery technologies to cooperate.

There are at least two interoperability approaches, namely *overlay* and *proxy*. The overlay defines a common layer, which hides the differences. This approach can be applied to many different SDPs simultaneously, while the proxy only enables interoperability between two SDPs. Various overlay solutions have been proposed, for example by Dobrev et al. (2002); Limam et al. (2007), introducing a new interoperability problem. Consequently, an overlay solution will gain popularity if it is implemented on widely accepted platform, such as OSGi. Alternatively, the proxy approach is suitable for less heterogeneous environments or when a point-to-point interoperability is desired. For example, a proxy can be used to extend service discovery in the home network to a personal area network.

Table 1.3: *SDP features comparison, derived from Meshkova et al. (2008).*

Technology	**Req.1**	**Req. 3**	**Req. 3**	**Req. 4**	**Req. 5**
	Network Transport	**Operation scope**	**QoS Support**	**Architecture**	**Standardized services**
Bluetooth SDP	Bluetooth Network	PAN	Yes	Distributed	Available
UPnP	IP	Small LAN	Yes	Distributed	Available
Bonjour	IP	Small LAN	No	Centralized	Limited available
Jini	Independent	Medium LAN	No	Centralized	Unavailable
Home Audio Video Interoperability (HAVi)	IEEE 1394	Small LAN	Yes	Centralized	Available but never in production
Salutation	Independent	Medium LAN	No	Centralized	Unavailable
Service Location Protocol	IP	LAN	No	Hybrid	Unavailable
JxTa	IP	Internet	No	Hybrid	Unavailable

1.4 Research Questions

The work in this thesis explores solutions for data communication problems in heterogeneous home networks. The present thesis work contributes research and several solutions in the field of network and service monitoring in home networks, mainly in three areas: (1) providing automatic device- and service-discovery and configuration, (2) remote management, and (3) providing quality of service (QoS).

As home networks are becoming ever more dynamic and technologically heterogeneous, the growth of these networks, in size as well as in number, leads to significant management complexities, for the users also for Internet service providers. Service Discovery Protocols (SDP) are an important component for mitigating these complexities. However, the existing SDPs are not platform and network independent. Consequently clients and services that are using different SDPs will not be able to cooperate. In addition, these SDPs hardly provide information on the actual availability of resources in the network or a mechanism for (device) resource reservation. When the resources cannot serve all multiple client requests at the same time, a conflict will happen. This conflict often involves heavy and frequent traffic reconfigurations. Therefore, in the context of addressing these SDP limitations, the following questions form

the basis of the research in this thesis.

1. How can we enable a seamless interaction between devices that use different network technologies and support different service discovery protocols, without adding more complexities to existing state-of-the-art?

2. How can we provide a generic solution for SDPs in providing information on the actual availability of (device) resources in the network so the resource conflicts can be avoided, without requiring any modification of current standards or devices in the home?

The raising popularity of IP-based- services has contributed to increasing complexity of home networks. This could be increasing service management difficulties for service providers, since the network of the devices behind the HG is beyond their control and they often consider home network as a large black box. A problem in the network would quickly lead to end users calling the service providers' help desks. Therefore, in the context of providing remote management of end-devices in heterogeneous home networks, the research in this thesis also addresses the following questions.

3. How can we provide a generic solution for remote discovery and management of end-devices in heterogeneous home networks, with minimal modifications to existing devices and remote management servers?

4. How can home network topology discovery protocols contribute to the service provider's remote management needs?

IP enables the interoperability of devices supporting different physical- and link-layer technologies and topologies. However, IP has only limited support for QoS, which is necessary to support many different services concurrently in a single shared home network. Many QoS solutions are available but they are still considered relatively expensive for mass-scale application today. Some novel approaches are under study, which are based on probing measurement and predicting some QoS parameters such as path capacity and available bandwidth. However, probing in home networks puts different requirements to a tool. In the context of estimating path capacity and available bandwidth in home networks, the research reported in this thesis is specifically aimed at answering the following questions:

5. How can we estimate path capacity and available bandwidth for heterogeneous home networks in a way that requires minimal modification to end-devices, which is not intrusive to the network, and is accurate enough for the most relevant home applications?

6. Is it possible to characterize home-network traffic and then build a prediction model for it?

1.5 Approach

The research reported in this thesis is aimed at answering the research questions formulated in Section 1.4. The research work was divided into several independent sub-investigations. In each sub-investigation, we performed the following phases:

- *Requirement Analysis.*
  In this phase, we performed a literature study for a theoretical context and existing systems. The result of this phase is the state-of-the-art and a general model of the problem domain. Complementary to this, we conducted a requirements study to identify design opportunities and additional user requirements.

- *Architecture and Implementation.*
  In this phase, we developed the architecture and the implementation of proposed concepts following the requirements set obtained in the previous phase. The activities in this phase show how the concepts are developed into a prototype, proof of concept, or test-bed.

- *Assessment.*
  This phase involves one or both of the following activities: (1) user testing of the prototype in a selected field setting and (2) running the test-bed in a simulation setting. The results of the assessment were analyzed and used to answer the research questions and to pose recommendations for further research and development.

In this research, we took five assumptions:

1. The home network is heterogeneous and configured as depicted in Figure 1.5. This configuration follows the definitions discussed in Section 1.1.4. Here, we added a new element, namely middleware, that is software implementing any SDP. In this research, we used Bluetooth SDP for SDP1 and UPnP for SDP2. In contrast to the project from which Figure 1.4 originate, this thesis does not discuss Peripheral Networks (PNs).

2. The home network has a single service provider configuring one HG. The HG is a powerful device such as a PC. The dots in the figure show properties of a device. For example, a device type of Retail End User Device 1 (R-EUD1) is a non-IP device, supporting SDP type 1 (SDP1), and has a home network (HN) type 1. The type of HN is distinguished by link-layer technology. In this research, we used Bluetooth network (*piconet*) for HN1 and Ethernet for HN2. HN3 is "no new wire" technologies that can be IEEE 802.11b, g, or Power-line technology.

3. The solutions and software proposed in this research are expected to run on the RG. The RG is our main solution space.

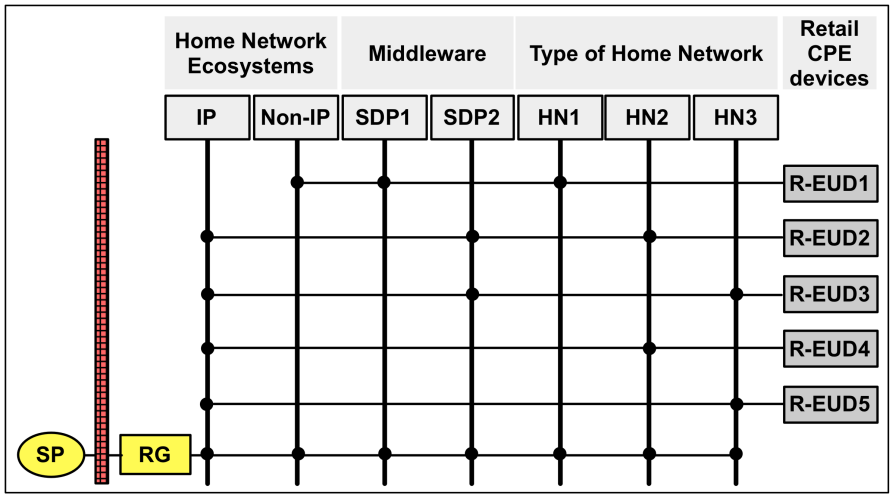


Figure 1.5: *Home network configuration in this research.*

4. In the context of topology discovery and of estimating path capacity and available bandwidth, the research only focuses on IP based end-devices.

5. If more than one device of the same type is needed, then instances of this device have the same processing power.

1.6 Thesis Overview and Contributions

After this introduction, the main chapters (Chapter 2-7) describe our work in answering the formulated research questions. Each chapter presents independent research, which has been published (in some cases as patent applications), as our contribution to the problem domain in focus. All chapters include their own introduction to the problem domain, background theories, proposed concepts, and experiments. They also present their own findings and conclusions.

This thesis is structured as follows.

Chapter 1 Introduction

This chapter introduces the research domain of home networking and, also, frames the research work presented in the thesis. It discusses the trends, technologies, challenges and opportunities in enabling interoperability home network. Based on the current state-of-the-art, then research questions are formulated and our approach toward solutions are outlined.

Chapter 2 A Proxy Server to Support Bidirectional Interoperability between UPnP and Bluetooth SDP

This chapter addresses research question no. 1. It presents the development and the evaluation of a proxy server. The proxy server enables UPnP devices and services to be discovered on Bluetooth network and vice versa. It allows Bluetooth devices and UPnP control points to access and utilize services located on devices in the other network. This work has been published in Delphinanto et al. (2007a,b) and patented in Delphinanto et al. (2009a).

Chapter 3 Integration of Resource Reservation with Service Discovery Protocol to Enhance Quality of Service

This chapter addresses research question no. 2. It presents the design, the implementation and the evaluation of a generic resource reservation scheme for avoiding resource conflicts in SDPs. The results have been published in Delphinanto et al. (2008).

Chapter 4 Remote Discovery and Management of End-User Devices in Heterogeneous Home Networks

This chapter addresses research question no. 3. It presents architectures for the unique remote discovery and management of devices in a highly heterogeneous home network. The architecture of the remote discovery has been adopted by the HGI. The chapter describes a proof-of-concept for the remote management of UPnP devices in the home with a TR-069/UPnP proxy on the HG. The findings of this chapter have been published in Delphinanto et al. (2009b).

Chapter 5 Path Capacity and Available Bandwidth Estimation for Heterogeneous Home Networks

This chapter addresses research question no. 5. It demonstrates and evaluates a new method to probe the path capacity and available bandwidth between a server and a client in a home network or any other best-effort small-scale IP networks. The software implementing the method has been demonstrated in the IEEE CCNC 2011 in USA and the Broadband-Home Seminar 2010 in the Netherlands. Moreover, it has been awarded as "the Best Innovation in Software Modularity and Applications for Home Gateway" by the Connected Home Global Summit 2011 in London, UK. The research work has been published in Delphinanto et al. (2010, 2011d,a,c), presented in den Hartog & Delphinanto (2010, 2011) , and patented in Delphinanto et al. (2011b).

Chapter 6 Path Analysis of Topology Discovery Protocols for Home Networks

This chapter addresses research question no. 4. It presents a set of key performance indicators for home-network topology discovery architectures, and demonstrates how they should be measured. The indicators can be utilized as

an evaluation framework, which can be applied to the Link-Layer Topology Discovery (LLTD) protocol and the Link-Layer Discovery Protocol (LLDP). The research results have been published in Castellanos et al. (2012).

Chapter 7 Traffic Prediction in Home Networks

This chapter addresses research question no. 6. It describes a method to model home-network traffic dynamics based on relatively few observations. The model can be used to make predictions of home-network traffic dynamics, for the purpose of network optimizations and QoS controls inside the home network. The findings of this chapter are currently under review for publication of Delphinanto et al. (2012).

Chapter 8 Final Remarks

This chapter highlights the main findings of our research. It thereby pertains to answer every formulated research questions. The chapter also points outs some directions for future developments.

Chapter 2

A Proxy Server to Support Bidirectional Interoperability between UPnP and Bluetooth SDP

The current service- and device- discovery protocols are not platform- and network-independent. Therefore, proxy servers will be needed to extend the range of IP-based discovery protocols to non-IP domains. In this chapter, we present an architecture of a proxy that enables UPnP devices and services to be discovered on the Bluetooth network and vice versa, and allows Bluetooth devices and UPnP control points to access and utilize services located on devices in the other network. Our analysis of resource requirement with a proxy prototype shows that all functionalities needed for effective proxying can be run on a single device, such as a mobile phone or a residential gateway. We also describe a design extending the proxy architecture to allow interworking between the UPnP Content Directory Service and the Bluetooth File Transfer Profile. Our performance study shows that our proxy implementation reduces invocation time and data throughput to about 50% of the bare Bluetooth and UPnP performance, but it is still acceptable for an end user. The findings of this chapter have been published in Delphinanto et al. (2007a,b) and patented in Delphinanto et al. (2009a).

2.1 Introduction

Service discovery technology is an important component for communication and service collaboration in distributed computing environments. In private networks, such as home networks and in-car networks, these environments are very heterogeneous considering the great variety of devices, network technologies, control protocols and application platforms being used. This is where auto-configuration protocols and device- and service discovery protocols come into play. These protocols (re)configure a home network dynamically, discover the device- and service properties to the other devices in the network. Examples of such protocols are Service Location Protocol (RFC2068), Jini (Jini.org, 2007), Universal Plug and Play (UPnP.org, 2008), and Bluetooth Service Discovery Protocol (SDP) (Bluetooth.org, 2007).

Unfortunately, these service- and device discovery protocols suffer from a number of limitations. The one relevant to this chapter is the following: none of the protocols is completely platform- and network independent. For example, UPnP operates at the network layer and above, and only runs on Internet Protocol (IP) networks. Bluetooth is a link layer protocol and Bluetooth SDP only runs on Bluetooth networks (and in the future probably also on Ultra-Wide-Band). As a result, Bluetooth devices such as headsets and some MP3 players cannot discover and use UPnP-enabled content servers in the IP part of a heterogeneous home network.

Most probably, time will not solve this problem. Although Bluetooth could support IP in principle, it does not do so in practice. Also home automation networks and in-car networks generally do not support IP, and probably will not in the mid-long future, because industry still considers IP as too resource intensive for small devices (headsets, sensors, etc.) and too vulnerable for critical communication such as car control. Therefore, proxy servers will be needed to extend the range of IP-based discovery protocols to non-IP domains.

This chapter presents an architecture and evaluates implementations of such a server. In general, the proxy server connects a (non-IP) Bluetooth *piconet* with a UPnP-enabled IP network, and provides proxy services on three different levels:

- It enables UPnP devices to be discovered on the Bluetooth network and vice versa.
- It enables UPnP services to be discovered on the Bluetooth network and vice versa.
- It allows Bluetooth devices and UPnP control points to control and whenever possible to access services located on devices in the IP network and the *piconet* respectively, independent of the type of service if possible. Some UPnP services may not be relevant for a Bluetooth client, for instance in the case of high quality video streaming. In this respect, the server is acting like an application gateway.

We set the proxy to fulfill the following requirements. First, all the functionality needed for effective proxying must be able to run on a single device such as a residential gateway (RG) and a mobile phone. Second, the proxy should also interoperate with as many other consumer electronics currently on the market as possible. Finally, the proxy server itself should require minimal configuration by the user. Correspondingly, these requirements allow the following benefits. It needs no modification on the current end devices in the market and owned by the user. Service providers can provide proxied services without having to synchronize with the consumer industry first. If the proxy runs on a mobile device with Bluetooth as the personal area network, the proxy can offer IP/UPnP connectivity anywhere.

This chapter also presents two evaluations of the proxy performance. The first evaluation is to study the resource requirement of the proxy, considering a PC-based implementation. The second evaluation is aimed at studying the interworking performance of the proxy. For the second evaluation, the proxy is implemented on a mobile platform and is used to allow a Bluetooth client to seamlessly access files in a UPnP server, which provides a UPnP Content Directory Service (CDS). The Bluetooth client sees and accesses the file service of the UPnP server as if it was a Bluetooth device supporting the Bluetooth File Transfer Profile.

The chapter is structured as follows. In Section 2.2, we discuss the state-of-the-art. Then, in Section 2.3 we give a brief overview of the UPnP and Bluetooth SDP frameworks. In Section 2.4, we continue with the architecture of the proxy and its resource requirement analysis. Further, in Section 2.5 the design and the proxy performance analysis for interworking between UPnP CDS and Bluetooth FTP are discussed. Finally, Section 2.6 draws the conclusions.

2.2 State-of-the-Art

Over the past years, a number of works have been conducted to enable the interaction between two different service discovery technologies. However, most of the previous works mainly focus on the proxy functionality, has a single direction of interoperation, and do not limit user interaction and resource requirements. Consequently, our proxy design requirements as discussed in Section 2.1, distinguish our work from the previous works.

The previous works typically proposed templates for protocol and data conversion. Koponen & Virtanen (2004) presented a proxy for Jini and Service Location Protocol (SLP) interoperability. The proxy allows all service advertisements mirrored in the directory service of each domain, which raises a scalability issue. The service access across the two domains however is not supported. Allard et al. (2003) presented a proxy architecture for Jini-UPnP interoperability. The proxy uses the client interfaces

from both SDPs to listen to any server announcements, generates a virtual server object for each discovered server and finally announces the virtual server to the counterpart network. This approach will also face scalability problem as all services of all domains are mirrored in each domain.

With a comparable architecture, Jun & Park (2004) proposed a unidirectional Bluetooth SDP-UPnP proxy server enabling a UPnP control point to control Bluetooth servers. They concentrate on proxying UPnP with Bluetooth SDP in a single device, but they only look in detail at the discovery of Bluetooth services by a UPnP control point, thus a single direction of interoperation, and do not limit user interaction. A different approach is found in the literature (Ayyagari et al., 2001; Atinav Inc., 2006; Bluetooth.com, 2001), that either focus on enabling UPnP to run on Bluetooth devices, or on applying an application layer overlay on top of Bluetooth SDP and UPnP stacks. This consequently requires extra (proprietary) software to run on all devices involved.

2.3 UPnP and Bluetooth SDP Overview

2.3.1 UPnP Basics

UPnP is an interoperability framework for devices and services in a relatively small-scale IP network. It is based on the client/server model and distinguishes three logical entities in the network: UPnP Services, which represent the service functionality of a device, UPnP Devices, which act as services servers, and UPnP Control Points (CPs), which act as clients for controlling the services. So a UPnP Device is not a physical entity, but a logical container of UPnP Services and embedded UPnP Devices.

UPnP defines protocols to provide the following functionality:

- Addressing, namely IP configuration of the network by using Dynamic Host Configuration Protocol (DHCP) or Auto-IP.
- Discovery of some key properties of a device and a pointer to its standardized XML device and service descriptions, by using Simple Service Discovery Protocol (SSDP). The information provided by the device descriptions is summarized in Figure 2.1(a). Service description contains service control information for the user interface of the CP.
- Control and Eventing of a UPnP Device or Service by using Simple Object Access Protocol (SOAP) and General Event Notification Architecture (GENA).

The SSDP defines how to find network services. It is both for control points to locate services on the network, as well as for devices to announce their availability. A

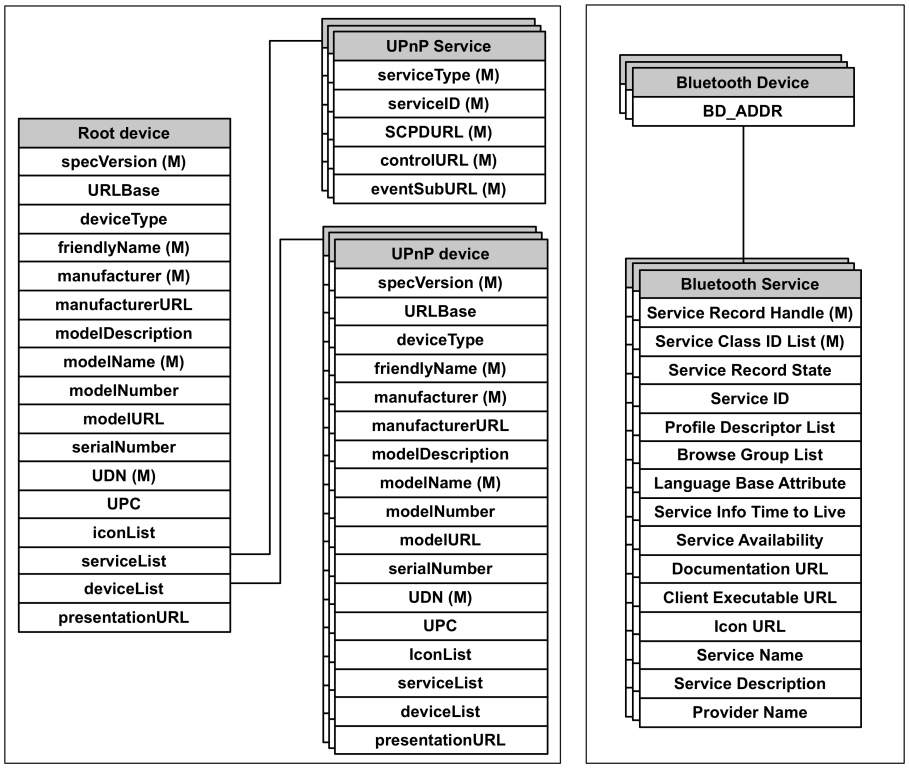


Figure 2.1: *(a) Overview of properties in a UPnP device description and (b) Bluetooth service record.*

UPnP control point, when booting up, can send a search request to discover devices and services. UPnP devices, on the other hand, listen on the multicast port. If the search criteria match, a unicast reply is sent. In the same way, when a device is plugged in to the network, it will send out multiple SSDP presence announcements. Apart from a leasing concept, SSDP provides a way for a device to notify that it is leaving the network. The leasing concept states the availability of a service being expired if after a preset time the service announcement is not renewed.

GENA defines the concept of subscribers and publishers of notifications. The GENA formats are used to create these announcements that are using SSDP. SOAP is then used to execute remote procedure calls, as well as to deliver control messages and return results or error messages to the control points. The device and service description, control messages, and eventing are expressed in eXtended Marked-up Languange (XML). The advertisement message that a device issues contains a URL that directs to an XML file in the network, which describes capabilities of the device.

UPnP forum, the organization that defines the UPnP architecture, also standardizes some devices and services such as UPnP Content Directory Service (CDS) for file transfer and browsing service. More standard devices and services can be found in (UPnP.org, 2008).

2.3.2 Bluetooth SDP Basics

Bluetooth is an industrial specification for wireless personal area networks. Primarily designed for low-power consumption and short-range coverage, Bluetooth provides a wireless way to connect and exchange information between mobile devices such as mobile phones, laptops, digital cameras and etc. A Bluetooth network is an ad-hoc network (*piconet*) that works in a master-slave fashion. A master may have connections to 7 active slaves and up to 255 inactive (parked) slaves. In Bluetooth communication, data can only be exchanged between a master and one slave at a time. However, these devices can switch roles and the slave can become the master at any time. Bluetooth link technologies (namely the Bluetooth core communication protocol) and applications (namely Bluetooth profiles) are maintained by Bluetooth Special Interest Group (SIG).

The bottom part of Figure 2.3 shows the relevant protocols of the Bluetooth core architecture. The SDP interfaces with the Logical Link Control and Adaptation Protocol (L2CAP), which is basically Bluetooth's link layer protocol. Via the common Host-to-Controller Interface (HCI), the lower layer protocols (called Bluetooth Controller, not shown) are addressed. Bluetooth also contains many higher layer protocols, but most of them are optional. A set of protocols needed to support a particular (collection) of applications is called a profile. Only SDP and Generic Access Profile (GAP) are mandatory.

The Bluetooth SDP is designed for highly dynamic communications. It only provides a means for client applications to browse available services or search for a desired service in a Bluetooth server. Each Bluetooth service is referred to as a Bluetooth service record and is an implementation of a Bluetooth service class. Each service class is distinct and is distinguished by a universally unique identity. To access the services, the Bluetooth SIG defines Bluetooth profiles, which describes standard operations and protocols for applications that use a Bluetooth communication interface.

The Bluetooth services delivery framework defines three stages: connection, service discovery, and description. Connection is performed by the Bluetooth Controller and consists of device discovery ("inquiry") and connecting ("paging") stages. A device is characterized by its Bluetooth hardware address. Service discovery and description is performed by SDP. SDP clients search for needed services based on specific attributes of those services or on the class of service. After discovery, the SDP enables browsing of services based on the retrieved service records. A service record contains

the attributes of a single service associated with the server. An example is given in Figure 2.1(b).

2.4 Proxy Server Design

2.4.1 Proxy Server Definition

A proxy server is a server that handles the requests of its clients by forwarding requests to other servers. A client sends a request for a file, control action, web page, or other resource to the proxy server. The proxy server provides the resource by connecting to the specified server and requesting the service on behalf of the client. A proxy server may optionally alter the client's request or the server's response, and sometimes it may serve the request without contacting the specified server. A well known proxy example is a web proxy, which is used for many purposes, such as enforcing acceptable network use policy, providing security, caching services, or reformatting web pages (e.g. into the ones that fit in cell phones or PDAs). In this work we are looking at proxies that are used to reformat device- and service discovery and control messages. In particular we focus on the example of forwarding a service access request of a non-IP Bluetooth client to any UPnP server. When a return is expected, the return is reformatted and forwarded back to the Bluetooth client.

The main challenge of this proxy is mapping these two very different architectures to each other (see Figure 2.1 to 2.3). The most important differences can be characterized as follows:

- Device discovery in UPnP is based on finding logical devices in the network, whereas Bluetooth is about finding physical devices.
- Service discovery in Bluetooth is reactive. In UPnP it can also be proactive, namely advertising services to the network.
- Service descriptions in UPnP and Bluetooth differ a lot.
- UPnP standardizes eventing, presenting, and controlling of services. In Bluetooth, services can be discovered using the Bluetooth SDP but have to be accessed using other protocols. It varies heavily between profiles.

The most important design requirement of the proxy is that all the functionality needed for effective proxying must be able to run on a single device, e.g. the RG or a mobile phone, thus avoiding any requirement on the end devices. This makes the proxy useful for the current end devices in the market and owned by the user. As such, service providers can provide proxied services without having to synchronize with the consumer electronics industry first. Preferably the proxy should run on a

mobile phone, if we assume that the Bluetooth network is the Personal Area Network of the user, wanting IP/UPnP connectivity at any place. In addition, we require that the proxy server itself needs minimal configuration by the user.

2.4.2 Approach

Figure 2.2 shows our approach with respect to device- and service discovery. The top part of the figure depicts the discovery of UPnP devices and services by a Bluetooth SDP client in the Bluetooth network. We assume that the proxy has only one Bluetooth hardware interface. Therefore the Bluetooth client device in the *piconet* will discover it as a single Bluetooth device, containing Bluetooth services 1A-6B that represent properties of the UPnP services 1-6 as well as the properties of the UPnP devices A and B that contain the UPnP services. The bottom part of Figure 2.2 shows the discovery of Bluetooth devices and services by a CP in the UPnP network. We assume that the proxy has only one IP address. Therefore a CP will discover it as a single UPnP root device. Because the proxy contains an SDP client, it can discover Bluetooth services actively by performing a search at set times.

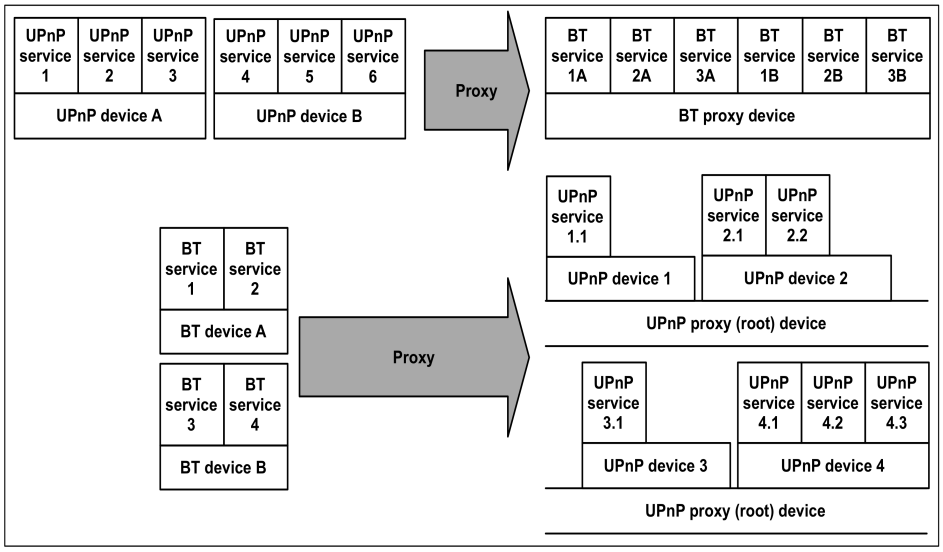


Figure 2.2: *The proxy translates devices as services in both directions.*

There are two options for representing the discovered Bluetooth devices and services in this root device. The most obvious one seems to have a UPnP embedded device representing a Bluetooth device, and its UPnP services to mirror the Bluetooth services. However, Bluetooth service records often contain much more information

than UPnP service descriptions (see Figure 2.1), whereas a Bluetooth device is identified with not much more than its hardware address. Also, when comparing the details of the descriptions and record of Figure 2.1, it can be concluded that it is easier to represent a Bluetooth service as a UPnP embedded device containing one or more UPnP services. The downside of this approach is that a CP does not know which Bluetooth device it addresses when invoking a service.

Furthermore, let us consider the following situation. There are two Bluetooth devices in the *piconet* offering the same service and with the same identifier (Service Record Handle). Then, a CP does not know which one to address, because Bluetooth devices are not converted into UPnP devices. Basically, the question is how the Unique Device Name (UDN) property of a UPnP embedded device is constructed from the Bluetooth service record. If the UDN is constructed from the Bluetooth Service Record Handle only, then this conflict cannot be avoided. However, Bluetooth SDP also specifies the universally unique Service ID attribute for the Bluetooth service record, which could be used as a UDN.

Unfortunately, the Service ID attribute is not mandatory in a service record. Therefore we suggest that the proxy builds a UDN by concatenating the Service Record Handle of the service with its Bluetooth hardware address. This also solves the problem of having two proxies in the same UPnP network discovering the same Bluetooth service: after discovering a Bluetooth service, a proxy should always check if it is not already offered in the UPnP network by the other proxy.

For many fields of the tables in Figure 2.1 the mapping is quite straightforward. For instance, the Bluetooth Provider Name corresponds to the UPnP Manufacturer, and the Bluetooth Icon List corresponds to the UPnP Icon URL. More examples can be found in Jun & Park (2004); Madureira (2006). Other fields are more difficult to translate, and can be left to the creativity of the proxy manufacturer or left blank. If left blank, only limited interoperability can be guaranteed. Standardization of this mapping by the UPnP Forum and the Bluetooth Special Interest Group would help to avoid large differences in interoperability between specific proxy server implementations.

For service presentation, invocation and control, the mapping is different for every service. The mapping of the related parameters as well as the service descriptions can be pre-programmed for the currently standardized Bluetooth profiles and UPnP devices and services. For new devices and services the proxy should be updated regularly, for instance by means of remote management by a service provider. Therefore to achieve seamless discovery of and access to current devices and services, a filtering module is needed. For example, the filtering module only accepts standardized UPnP that correspond to any Bluetooth profile offering similar functionalities. In practice, a service provider can configure and upgrade this filtering function via remote management when new UPnP and Bluetooth standards are released.

2.4.3 Architecture

Figure 2.3 shows the main components of the UPnP-Bluetooth proxy. Besides the UPnP and Bluetooth protocol stacks (white), the core architecture (grey) contains a *device and service discoverer*, a *converter*, a *device and service announcer*, a *UPnP adapter*, a *Bluetooth adapter*, and a *database* consisting of various tables. They are explained in more detail below.

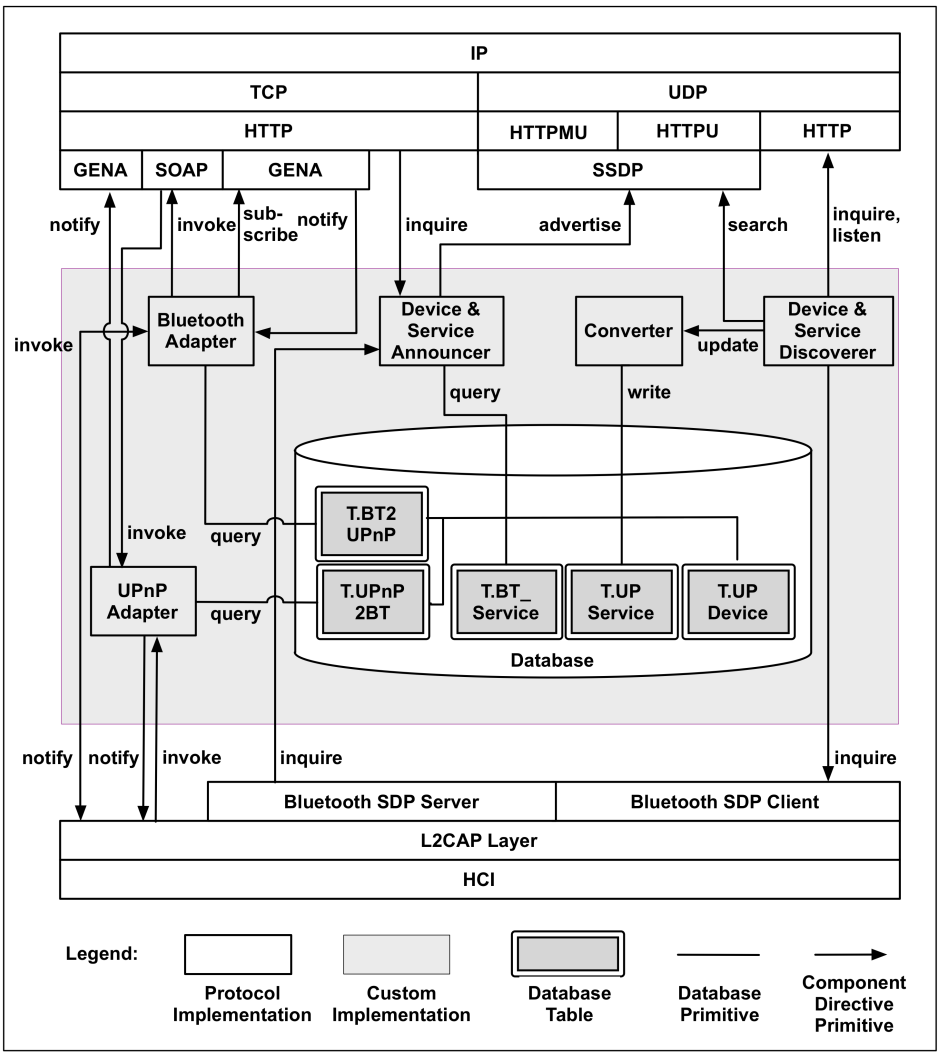


Figure 2.3: *Architecture of a bidirectional Bluetooth-UPnP proxy.*

- *Device- and service discoverer.* The device- and service discoverer is responsible for gathering the discovery information from the UPnP and Bluetooth stacks and passing it to the converter. It listens passively to the information provided, but also triggers active searches at reasonable time frames. It only passes information to the converter that actually has a corresponding service on the other side of the proxy. It also tells the converter when a service is unavailable after it detected that a Bluetooth link is disconnected, when the service is expired, or after receiving a UPnP `SSDP:byebye` message.

- *Converter.* The converter performs the mapping of descriptions, presentation, invocation and control, as described in the previous section, and writes the results to the database. It will also remove the relevant information from the database whenever a service has become unavailable.

- *Device- and service announcer.* The announcer advertises a newly discovered Bluetooth service to the UPnP network after an inquiry by a CP. It can also advertise spontaneously whenever a new Bluetooth service has been found. However, we advise to schedule these spontaneous advertisements at specific time intervals only, e.g. 60 s, to hide the dynamics of the Bluetooth network from the UPnP network and limit overhead traffic.

- *Database.* The database consists of five tables. The table `T.UPdevice` contains the UPnP device description of converted Bluetooth services. The table `T.UPservice` contains the corresponding UPnP service descriptions. The table `T.BTservice` contains the Bluetooth service records of converted UPnP devices and services. `T.BT2UPnP` and `T.UPnP2BT` respectively contain the information that is needed for a Bluetooth client to invoke any action of a UPnP service, and for a CP to invoke any action of a Bluetooth service.

- *Bluetooth adapter.* This component is basically an implementation of a CP that is controlled by a Bluetooth device in the *piconet*. This means that it listens to any Bluetooth client's invocation for a UPnP service and to any Bluetooth client's inquiry for UPnP service state variables. It can subscribe to any UPnP service's event, and it consequently receives notifications when UPnP service variables change. All information that the Bluetooth clients need to control the Bluetooth adapter is given by `T.BT2UPnP`.

- *UPnP adapter.* This component is basically a Bluetooth client controlling other Bluetooth devices in the *piconet*, and being controlled by a CP in the UPnP network. It listens to any UPnP control point's invocation of a Bluetooth service, and it notifies its service subscribers of any variable that has changed. Whenever a CP requires, it sets up an active connection to the appropriate Bluetooth

device and activates the relevant Bluetooth profile or custom application. In case of a headset this connection will probably not use L2CAP as shown in the figure, but will use the Bluetooth Controller directly (Bluetooth.org, 2007) All information that CPs need to control the UPnP adapter can be found in table `T.UPnP2BT`.

2.4.4 Analysis of Resource Requirements

To investigate the resource requirements of the proxy, we have implemented a part of the architecture on a PC (Intel Pentium 4 processor (1.6GHz), 512MB RAM, Fedora Linux OS) for a small number of UPnP and Bluetooth services. We measured the time taken for a Bluetooth service to be discovered by a CP, for a UPnP service to be discovered by a Bluetooth device as a function of CPU load. We also measured end-to-end data throughput when either service is invoked as a function of processor speed.

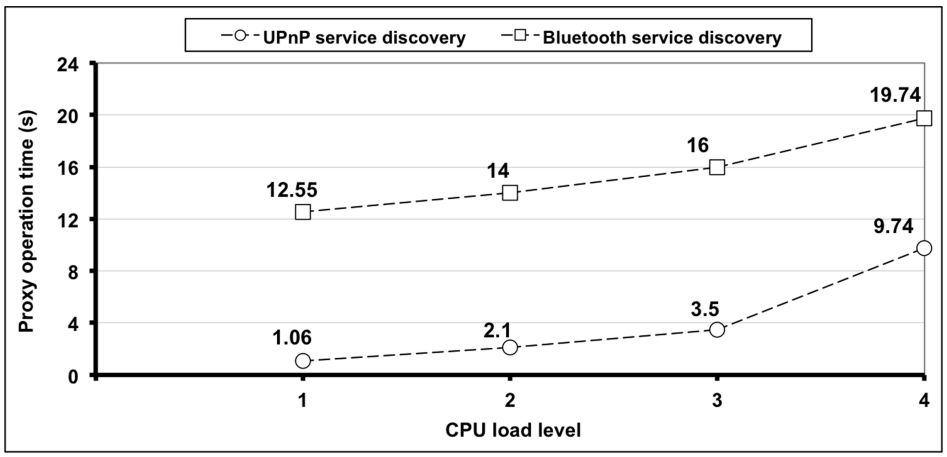


Figure 2.4: *Proxied discovery times as a function of CPU load (qualitatively).*

Figure 2.4 shows the discovery times as a function of CPU load. Unfortunately, the CPU load could only be measured qualitatively. Level 1 means that the CPU is basically used for the proxy operations only. Level 4 means that the CPU is used by other applications for more than 90%. The other levels fall in between more or less linearly. The Bluetooth service discovery time (square) is the time that a Bluetooth device needs to find a UPnP service. Discovery times in the order of seconds are also known for native UPnP systems. The UPnP service discovery time (diamond) shows the time that a CP needs to find a Bluetooth service. It shows that the Bluetooth discovery time is always much higher than the UPnP discovery time. This can be explained by the inquiry time that the Bluetooth Controller needs to find a Bluetooth device. In-

quiry times of more than 10 s are consistent with the literature for native Bluetooth systems (Madureira, 2006). At higher CPU loads, the UPnP discovery time increases much faster than the Bluetooth discovery time. A possible explanation is the fact that UPnP is an application-layer protocol and demands much more from the CPU than Bluetooth, which is basically a link-layer protocol that runs on the network interface card. From the figure it can be concluded that there should not be too many applications running besides the proxy software, because discovery times of over 20 s are hardly acceptable to the user.

We found that the end-to-end throughput was always limited by the throughput of the Bluetooth network rather than by the proxy. Furthermore we found that none of the performance parameters showed any dependence on the processor speed (varied between 600 and 1600 MHz). Finally we determined that the maximum RAM memory needed to run the proxy was 72 MB, which was required by the UPnP library for the discovery of UPnP services. We can therefore conclude that the proxy can run on a device with a processor speed of 600 MHz or less, and with 72 MB RAM or less, given the specific UPnP and Bluetooth implementations that we used. These requirements are met by most up-market mobile devices and RGs today. Most probably also much smaller devices will be able to run the proxy software, using a more dedicated implementation than ours.

2.5 Interworking between Bluetooth FTP and UPnP CDS

2.5.1 UPnP CDS Basics and Bluetooth FTP Basics

The UPnP standard services used for testing the interworking between Bluetooth SDP and UPnP is UPnP Content Directory service (CDS) (Presser et al., 2006). This service is aimed to provide a uniform mechanism for a user interface application to browse, locate, or transfer any content (for example, songs or video files) from a server and to obtain detailed information about individual content objects. Beside the mechanism, the commands for running the CDS are also standardized. Some example commands of the CDS are `browse()`, `importResource()`, and `search()`. For file transfer the CDS uses existing transport protocols namely `HTTP GET` (for downloading) and `HTTP POST` (for uploading).

The Bluetooth profile that corresponds with UPnP CDS is Bluetooth File Transfer Profile (FTP) (Gratton, 2002). The profile requires the devices to follow the Bluetooth Generic Object Exchange Profile (GOEP) and uses OBject EXchange (OBEX) as the file transport protocol. It enables a client to browse, transfer and delete files in a server. Those commands are carried out by combinations of OBEX commands (`CONNECT`, `DISCONNECT`, `PUT`, `GET`, `SETPATH` and `ABORT`). For example, the client browses files

in the server using `GET FolderListing` and sets a current folder using `SETPATH`, or the client may retrieve or upload a file using `GET` and `PUT`.

2.5.2 Interworking Design

To explain the proxy functionality for interworking between the Bluetooth FTP and UPnP CDS we developed a scenario. In a home, a user with a Bluetooth portable MP3 player wants to download a music file from a UPnP server. This activity cannot be done because the MP3 player has only a Bluetooth interface, the server has only a IP interface, and the Bluetooth applications cannot directly communicate with UPnP applications. The user has a mobile pocket PC that has Bluetooth and Wifi interfaces and runs the proxy application. Using this pocket PC as an intermediary, the user manages to download the file.

The implementation of the scenario is explained using a sequence diagram (depicted in Figure 2.5). Here the device and service discovery is assumed to have been carried out using the technique described in Section 2.4. The sequence diagram contains a client, representing the Bluetooth MP3 player; a server, representing the UPnP server; and the proxy. The actions in the scenario can be grouped into the following technical activities (distinguished by different time slots in Figure 2.5): (1) initialization, (2) initial browsing (a file structure), (3) browsing folder information, and (4) downloading (a file).

- *Initialization.* Before starting the other activities, an initialization is carried out. The server that contains a UPnP CDS should be discovered by the proxy, for instance by sending a `notify()` message. Then, the proxy should perform a UPnP to Bluetooth service static description mapping, for instance as described in (Jun & Park, 2004). Upon receiving the server's notification, the proxy invokes the server with a `browse()` action. The server will reply with an XML file (A1 in Figure 2.5) containing folder (root and its children) information. This information is used as an initial reference to the folder locations. Simultaneously, the client discovers the proxy and establishes a connection, and subsequently the client looks for a Bluetooth FTP.

- *Initial Browsing.* To obtain a folder hierarchy (root and its children) in the server, first the client invokes the proxy using a `GET` action with a `FolderListing` parameter. To respond to the invocation, the proxy will first reformat the file A1 into another XML file (B1) and send this file to the client. The file reformatting is needed because the UPnP XML file has a different structure and meaning from the Bluetooth XML file. For example, UPnP identifies folders by an identity while Bluetooth (OBEX) identifies folders by a name. For this reason, the proxy should keep pairs of Bluetooth folder names and UPnP folder identities.

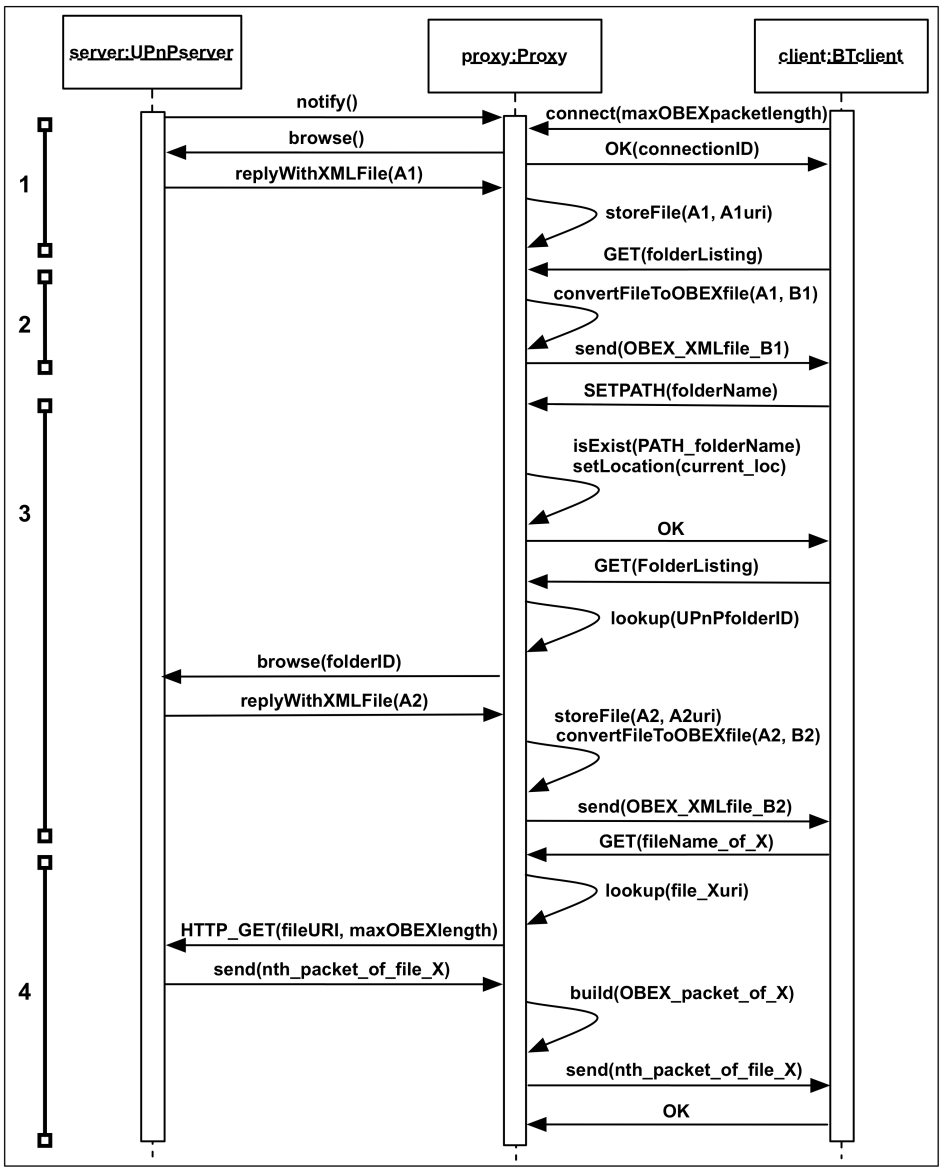


Figure 2.5: *Sequence diagram for interworking between UPnP CDS service and Bluetooth client.*

When the Bluetooth client has received the file B1, the client may parse it to a user-friendly format for the end user. The user can browse through the folder like any other file browser he is used to.

- *Browsing Folder*. To change a folder hierarchy, the Bluetooth client invokes the proxy with a `SETPATH` action with a folder name as parameter. The proxy will match the given folder name with available folder identities and in case of successful matching, the proxy replies the invocation with an `ok` message. To get the final folder structure, the Bluetooth client invokes the proxy again using a `GET(FolderListing)`. Accordingly, the proxy invokes the server with a `browse()` action, with the matched folder identity as parameter. The server will respond to the proxy's invocation with a new XML file (A2) containing the new structure information of the requested folder. Similarly, like in the browsing, A2 is converted to a new OBEX file (B2) and sent to the client.

- *Downloading*. The client downloads a file from the server by sending a `GET` action to the proxy with the desired file name (X) as parameter. The proxy looks up the location (namely file-uri) of file X. Then it invokes the file by sending an `HTTP GET` to the X's uri with the maximum OBEX packet size (obtained in the initialization) as parameter. The server will send file X to the proxy in several packets depending on the given OBEX packet size. The proxy wraps each of these packets into a Bluetooth OBEX packet and sends it to the client. Upon each successful packet reception, the client must confirm to the proxy with an ok message, after which the proxy can continue downloading the file. This is repeated until the file is completely transferred. The proxy acknowledges the client about the completed transfer by sending a download-finished message.

During the implementation, we found that the UPnP CDS and Bluetooth FTP work quite differently, though they have similar functionality and commands. For example, for browsing, the client needs to send two invocations (namely `SETPATH` and `GET`) while the server only needs one invocation (namely `browse()`). From this we conclude that the proxy needs specific solutions for each conversion between Bluetooth profiles and UPnP services.

2.5.3 Performance Analysis

To validate our current architecture, we created a scenario and implemented the concept in a mobile platform. In this section, we focus on the service access functionality of the proxy and the respective performance analysis. Our implementation consist of a mobile platform (HTC TyTN, 400 MHz processor, 64 MB RAM, Bluetooth v2.1 and IEEE 802.11b interfaces), an operating system (Microsoft Windows Mobile 5.0), an internal database (Microsoft SQL CE Server), a UPnP library (Intel UPnP CE Stack), a Bluetooth library (32Feet Bluetooth Stack) and an XML Parser library (System.XML library provided by the .Net framework).

Using the implementation, we measured the proxy's response times for browsing files and downloading a file. For the browsing response time we measured the total browsing time and the UPnP browse time as perceived by the proxy. The total browsing time is measured from the time the proxy receives the client's request (namely `SETPATH`) until the time the proxy replies the `Get(FolderListing)` to the client. This total time is identical with the time slot number 3 in Figure 2.5 (namely browsing folder). The total browse time minus the UPnP browse time yields how much latency the proxy added to the original UPnP browse time. The downloading time is measured from the time the proxy receives a client download request (`GET` file) until the time the proxy sends the download finish acknowledgment to the client. Each measurement is repeated 40 times to minimize sampling errors.

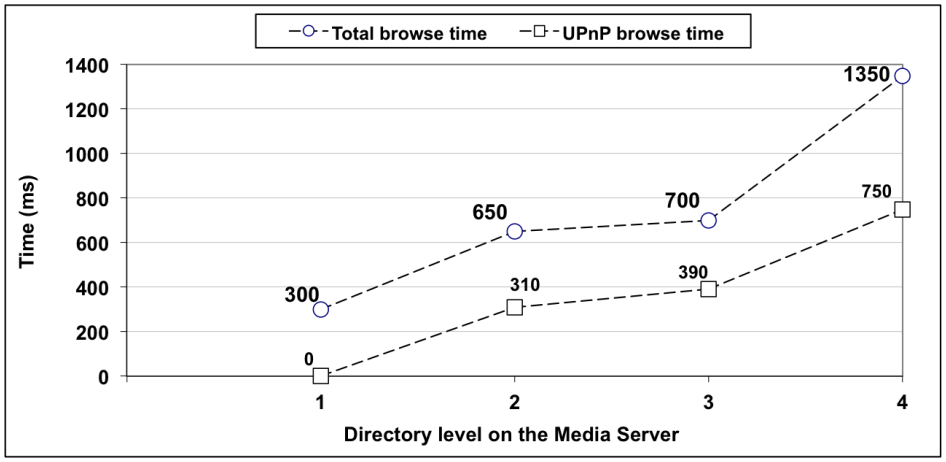


Figure 2.6: *The proxy browse time at different levels of file folder.*

Figure 2.6 shows the measured total browse time and UPnP browse time as a function of folder level (namely the position in the folder hierarchy). Level 1 represents a browsing to the root (top) folder hierarchy of the file server. In our case the root contains three folders. The root information is obtained during the initialization process (as discussed in Section 2.5.1). Therefore, the root browsing does not require a separate server invocation. This explains why the UPnP browse time at level 1 is zero. Level 2 represents browsing to one of the three subfolders of the root folders. Level 3 means browsing to one of eleven subfolders of the level 2's folders. Finally, level 4 shows the browsing to the lowest position of the folder hierarchy, which contains 28 files. The figure shows that the total browse time is almost double the UPnP browse time and both response times increase with the amount of folder/file information available. From this we can conclude that the proxy processes for the browsing

activities (namely looking up folder/file identities, updating new folder information and converting UPnP XML folder structures to OBEX folder structures) take about half the total browsing time, which is significant. However, also for complicated file folder structures (e.g. with four levels), the total browsing time is still acceptable for the end user. From this we conclude that our proxy design can indeed be implemented on current mobile platforms, adding value to current Bluetooth and UPnP services.

We also measured the payload data throughput when downloading a file from a UPnP server to a Bluetooth client via the proxy. We found a throughput of 150 ± 30 kbit/s, which is about half the native Bluetooth throughput. The throughput on the UPnP network is faster than on the Bluetooth network (relatively around 7 Mbit/s), so the loss can be accounted to the operations of the proxy. For some applications such as streaming audio, the measured rate is still high enough.

We also repeated the measurement adding a second Bluetooth FTP client that downloads another file simultaneously during the download time measurement. We then obtained a rate of 70 ± 20 kbit/s. This rate drop occurs because the Bluetooth link fairly divides the number of channels between the two clients. In this case the audio file has to be downloaded or partly buffered before being consumed to achieve an acceptable audio quality. From this we conclude that the number of Bluetooth links to the proxy should be limited, namely two Bluetooth links.

2.6 Conclusions

Proxy servers will be needed to extend the range of service- and device discovery protocols to non-IP domains, such as the Personal Area Network or the car network. This chapter presents and discusses an architecture of such a server. It connects a (non-IP) Bluetooth *piconet* with a UPnP-enabled IP network, it enables UPnP devices and services to be discovered on the Bluetooth network and vice versa, and it allows Bluetooth devices and UPnP control points to control service located on devices in the IP network and the *piconet*, respectively. Although the architecture looks fairly straightforward, this is the first time that it has been put in practice for evaluation.

The most important property of our architecture is that all functionality needed for effective proxying is kept within a single physical device. There are no additional stacks or software needed on the other devices in the *piconet* (or IP network). This enables the discovery of not only complex devices, but also very simple ones, such as headsets and toys. It also allows service providers to provide proxied services without having to synchronize with the consumer electronics industry first. The proxy can be configured remotely by service providers and therefore requires minimal configuration by the user.

To reach a maximum interoperability, the proxy is required to run all Bluetooth

profiles and UPnP standard implementations that currently exist. New profiles, standards, and converter software could be provided by means of remote management. But even then a number of standard Bluetooth services can not have clearly corresponding UPnP services, and vice versa. As a consequence, the match between Bluetooth profiles and UPnP standards is sometimes impossible, or only possible for part of the functionality. We also found that the UPnP CDS and Bluetooth FTP work quite differently, though they have similar functionality and commands. Therefore the proxy needs specific solutions for each conversion between Bluetooth profiles and UPnP services.

From the resource requirement investigation, it can be concluded that that the proxy software can run on current standard mobile devices and RGs as long as other applications are not competing too much for processor time. Furthermore, from the performance evaluation, we concluded that the Bluetooth part always becomes the end-to-end bottleneck, the proxy delays the access (namely browse) time and reduces data throughput to about 50% of the bare Bluetooth and UPnP performance, but this is still acceptable for an end user.

Chapter 3

Integration of Resource Reservation with Service Discovery Protocol to Enhance Quality of Service

Current service discovery protocols (SDPs) hardly provide information on the actual availability of resources in the network or a mechanism for (device) resource reservation. When the resources cannot serve all multiple client requests at the same time, conflicts happen, often involving heavy and frequent reconfiguration traffic. In this chapter, we present a generic resource reservation scheme that is useful for SDPs. Its properties are derived from common SDP operations. We then describe its implementation on UPnP, simulate its gain in network scalability, and evaluate its extra overhead. The main conclusion is that our solution improves the scalability and the sustainability of the service access significantly, and at a minor cost. The findings of this chapter have been published in Delphinanto et al. (2008).

3.1 Introduction

Service Discovery Protocols (SDP) (re)configure a private network dynamically, discover the device- and service properties of the devices present, and communicate these properties to the other devices in the network. Current protocols are still relatively immature and suffer a number of limitations. The one relevant to this chapter is the lack of information on the actual availability of resources in the network or a

mechanism for (device) resource reservation.

Device resource reservation is required whenever a device acts as a server in the network and shares its services to multiple clients but, because of its limited resources, can only serve a limited number of clients simultaneously. For example, most wireless music receivers that are on the market today can only provide audio streams to one client at a time. Another example, UPnP specification for media renderer, namely MediaRenderer:1 (Ritchie, 2002), does not provide the availability information. Devices implementing this specification do not report their current state as being occupied and, often, they still accept service access requests by other clients. This leads to overriding of the first client, to a strong increase of unnecessary configuration traffic because, for instance, the first client may try to regain the service access, or to other erroneous behavior. This problem has been mentioned in the literature before (Yamazaki et al., 2005; Lea et al., 2000). What is needed is service access management to solve the conflict, or more generic, resource reservation management.

This chapter presents and discusses a concept of a resource reservation manager. It enables a client to reserve relevant service resources for sustainable service access and protect this reservation from other clients. Three important design requirements were applied. The reservation manager should be:

1. based on a centralized architecture. Hence, it can gracefully be implemented on the existing SDPs and will avoid extra requirements (and therefore costs) on the current server specifications;

2. independent of specific SDP properties that are not shared by other SDPs. This makes bridging easier to private networks that are governed by other SDPs;

3. manageable, possibly remotely, so it can be updated with, for instance, more advanced policies.

The resource reservation manager is implemented by using Universal Plug and Play (UPnP) UPnP.org (2008), where the overhead it introduces is measured and the gain in network scalability is simulated.

The chapter is structured as follows. In Section 3.2, we discuss the relevant properties of SDPs and the state-of-the-art on resource reservation management. We continue in Section 3.3 by showing how much unnecessary traffic may be generated by such a conflict. The next is our reservation manager design is elaborated in Section 3.4, followed by a description of our reservation manager implementation on a UPnP platform in Section 3.5. Further, the performance of the implementation in terms of scalability and overhead is presented in Section 3.6. Finally, we give some conclusions in Section 3.7.

3.2 SDPs and Resource Reservation

Service discovery technologies have been developed to reduce the complexity involving the manual configuration of multiple devices to form a local network, even if the network is still relatively small. These technologies enable users to automatically discover network services, configure their properties, and access their functionalities. Results of various surveys on these technologies (Bettstetter & Renner, 2000; Zhu et al., 2005; Richard, 2000) conclude that these protocols contain mostly the same elements and functionalities but have different approaches, and that none of these technologies is superior. The elements and functionalities of the SDPs that are similar and useful to this chapter are:

- SDP elements always contain clients (service users), servers (service containers) and a service-repository. In distributed architectures, such as UPnP, the repository's job resides in the server.
- The SDPs contain a uniform grammar and semantics of service and device descriptions that enable a client to learn a newly deployed server without prior knowledge of the server, and then to use its services.
- Automated service discovery is done by announcement (except Bluetooth SDP (Bluetooth.org, 2007)) and query (Zhu et al., 2005). An announcement lets the server advertise its services to the network and a query lets the client actively discover a desired server.
- Dynamic server information (state or variable change) is updated to any clients with event subscription and notification.
- The server lifetime is leased and should be renewed periodically (not available in Salutation (Bettstetter & Renner, 2000)). This prevents deadlocks and inconsistent server states whenever the server has been unintentionally out of service.

Among the SDPs mentioned, Home Audio Video Interoperability (HAVi) (Havi.org, 2001) is the only one that provides a form of resource management. HAVi is a relatively advanced SDP designed especially for audio and video related devices. The resource manager lets clients reserve resources, release resources and arrange scheduled actions. These scheduled actions identify a time when the resource is reserved, and can be appointed to the interested clients to prevent potential conflicts. Reserved resources may be preempted by a negotiation with the resource owner, which may be a person. Resource reservation by human intervention has always priority over automatic reservation by the system.

Unfortunately, the HAVi resource reservation concept cannot be easily adapted to other SDPs. Other SDPs mostly rely on the Internet Protocol (IP), whereas HAVi is built

on the IEEE 1394 bus standard. Consequently, services based on other SDPs cannot participate in HAVi resource reservation. Furthermore, HAVi resource management works in a distributed fashion. It requires an extra resource management software element on all participating devices in the network, thus introducing problems in terms of backward compatibility and upgradeability, and additional costs to every single device.

3.3 Traffic Generation by Service Access Conflict

A significant consequence of service-access conflict is the generation of unnecessary traffic. To quantify this effect, we made an analysis of the conflict, assuming a network that consists of a server and multiple clients. A conflict arises when the clients compete for access to the server. To ease the analysis, some assumptions are made:

- All clients, which access the server will generate configuration packets that consist of controlling and response messages.
- The clients' requests arrive at the server at random times $(t_1, t_2 \dots t_i)$, represented as random time functions T_i.
- The server does not have a resource manager. Therefore, server access granted to a particular client can easily be interrupted by any other client.
- Without awareness of the conflict, each client will always retry to regain the server session after a timeout period T_o.

The first conflict happens when a second client sends an access request at time t_2 while the first client has not finished using the server. As a result, the first client will lose the server session and try to re-access the server by sending another access request after T_o. Later, before the first client re-accesses the server, the third client has sent an access-request and replaced the second client's server session. Similarly to the first client, the second client will also retry to re-access the server after T_o. The conflict will continue and become worse with increasing number of clients.

$$Traffic = [n + \sum_{i=2}^{n} int \frac{T_s - T(i)}{T_0}] \cdot CP \tag{3.1}$$

The total traffic generation because of the conflict during a server lifetime T_s is given in Equation 3.1, which can be obtained from the total number of incidents at which the clients send configuration packets, multiplied by the configuration packet size (CP). The first term shows the initial configuration traffic because each client sends a configuration packet for the first time. It corresponds with the number of

clients n. The following term shows the unnecessary overhead traffic because each client will try to get server access by sending a configuration packet periodically at an interval of T_o until the end of T_s. Equation 3.1 assumes that there is more than one client in the conflict, that all the clients have the same T_o, and that CP is the same for each client and request.

3.4 The Resource Reservation Manager

In this section, we explain our proposed resource reservation manager for service discovery protocols that fulfill the requirements given in Section 3.2.

3.4.1 Concept

The basic operation of the reservation manager is shown in Figure 3.1. The network consists of a client, a server and the reservation manager. The solid lines represent the physical connection to the network and the dashed lines show the exchange of messages. First, the server registers its services to the reservation manager and the clients, which is done using the SDP's native advertisement mechanism. Upon receiving the information, the reservation manager registers/updates the (new) services in its local service registry database. Each service registry includes, amongst others, universally unique identification, service state (explained in Section 3.4.2), service user, service lifetime, and user reservation time (namely how long the service will be engaged by the service user).

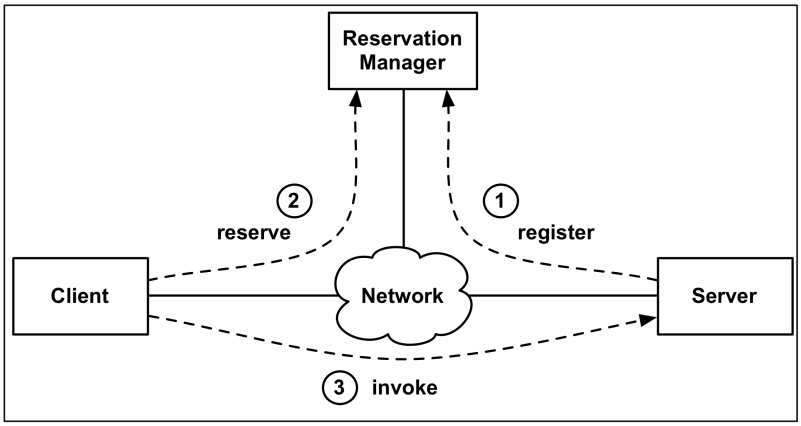


Figure 3.1: *Basic operation of the reservation manager.*

Then, the client will query the reservation manager to reserve access to the ser-

vice for a given duration. The reservation manager will check the availability of the requested service and whether it is possible to be reserved, which depends on the reservation policy present on the reservation manager. In case the service can actually be reserved, the reservation manager will respond the request with a "successful reservation" message, which also contains the duration of the reservation. Otherwise it will return a "failed reservation" message, which may contain a duration after which access can be retried. Additionally, the client may want to extend the reservation. For this, the client should reserve the service again before the reservation expires. This is the same mechanism as the leasing concept mentioned in Section 3.2.

Finally, the client may stop using the service before the reservation ends. The client then must release its service reservation. There are two ways for this. First, the client sends "release reservation" to the reservation manager. The manager then changes the reservation state and informs the other potential clients. Thus the other potential clients do not need to wait until the suggested retrial time for reserving the service. Second, the service reservation of the client will finish by itself after the reservation duration has ended. Because the client does not renew the reservation, the service state automatically becomes available.

3.4.2 Service State Diagram

To provide the clients with useful service state information, a service state diagram is proposed as depicted in Figure 3.2. The state diagram shows different service states and relevant transitions. A service is *Unavailable* when it is known by the reservation manager but not ready for accepting any jobs. A service becomes *Ready* when it is known by the reservation manager and all clients in the network and is ready for accepting any jobs. Finally, a service is *Reserved* when it is known by the reservation manager and all clients in the network, but its access is only granted to the client who has reserved it.

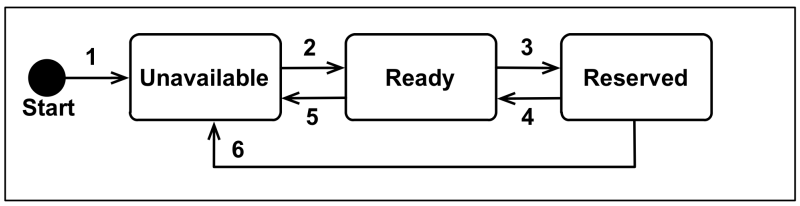


Figure 3.2: *A simple service state diagram.*

A new service is *Unavailable* from the *Start* (1). The service will become *Ready* (2), if its server has advertized it to the network. This will change to *Reserved* (3), if there is a successful reservation by any client. The service returns to *Ready* (4), if the client

ends the reservation or it does not renew the reservation. Alternatively, the reserved or ready service can become *Unavailable* (5 and 6), if the server becomes inactive intentionally or accidentally.

3.4.3 Reservation Policy

Let's first assume that the reservation policy is based on a fair context, namely "first come first served". The earlier a client requests a reservation, the more priority it gets. Consequently, the later incoming clients will be put in a queue and are suggested a retrial time by the reservation manager. The retrial time T_{ret}, of a client is given in Equation 3.2 as a function of the client's position in the queue (i), and is an accumulation of earlier clients' reservation times T_{res}.

$$T_{ret}(i) = \sum_{k=1}^{i-1} T_{res}(k) \tag{3.2}$$

As each client i may reserve the service for a different duration, the reservation manager should keep these individual durations with the position of the clients in the queue. Furthermore, it is likely that any client may want to continue its service reservation before the current reservation ends. Then the client should renew the reservation before the reservation gets expired. Alternatively, the client may release the reservation. For those two conditions, the reservation manager must update the queue table and disseminate the new information to the other clients.

Including this reservation management, the total configuration traffic until the last client accesses the server can be predicted by Equation 3.3. The total configuration traffic is now given by the number of clients n each generating one configuration packet (CP) to access the server and one reservation packet (RP) to reserve the service (including the response), and additionally m announcement packets (AP) that are generated if any client stops or extends the reservation before the reservation gets expired.

$$Traffic = n \cdot (RP + CP) + m \cdot AP \tag{3.3}$$

The reservation policy can be easily extended with any other more advanced contexts such as bandwidth availability (Giovanelli, 2003) and identity of the client. Of course this will then affect Equation 3.2 and 3.3.

3.4.4 Discussion

A reservation manager as defined above does fulfill the requirements given in Section 3.1. Requirement 1 is fulfilled when the reservation manager, as depicted in Figure 3.1,

is based on a centralized architecture and makes use of native SDP messages only. This is the most innovative part of this work. We do not impose modification of any standard servers (devices). The reservation manager can be added to an existing SDP network as a new service without any additional software or configuration of the current server specifications. However, we do require a modification in the client behavior. To have effective and consistent service-access reservation, every client in the network needs to request a service reservation from the reservation manager prior to accessing the service. The client must refrain itself from accessing the service whenever the requested service is reserved by another client. To eliminate this client requirement, the reservation manager may be combined with the authentication mechanism provided by the SDP, like the UPnP *security ceremony* (Elisson, 2003). Only clients that have been granted permission can then access the reserved service. This permission should then not be based on identity only, but also on other client properties such as the ability to request and release reservations.

Requirement 2 is fulfilled for all SDPs that support the functions as bulleted in Section 3.2. As mentioned there, Bluetooth SDP does not define any advertisement protocol. To monitor service availability anyway, the reservation manager should then periodically invoke the server. This is a fairly complicated and bandwidth consuming work around though.

The third requirement is fulfilled if the reservation manager is implemented in a remotely manageable device, such as a residential gateway (den Hartog et al., 2004; HGI, 2006) or a mobile phone.

3.5 Implementation on UPnP

Figure 3.3 shows the architecture of our reservation manager implementation on UPnP. The architecture is composed by four modules, namely a Service Discovery module, a Database, a Decision Engine, and a User Interface module. The modules function in the following manner.

- *Service Discovery.* This module is responsible for monitoring service advertisements and for listening to any client request for service reservation. This module contains all basic UPnP elements, namely a CPt and a server incorporating a reservation manager Device with a corresponding Service. In our implementation, this module is developed using a Java UPnP toolkit from Cyberlink (Konno, 2008). The CPt will discover any server in the network using standard service discovery operations (namely search and listen) and then registers it to the database. The Device periodically advertises the reservation manager Service to the network, and it listens to any service invocation. The reservation

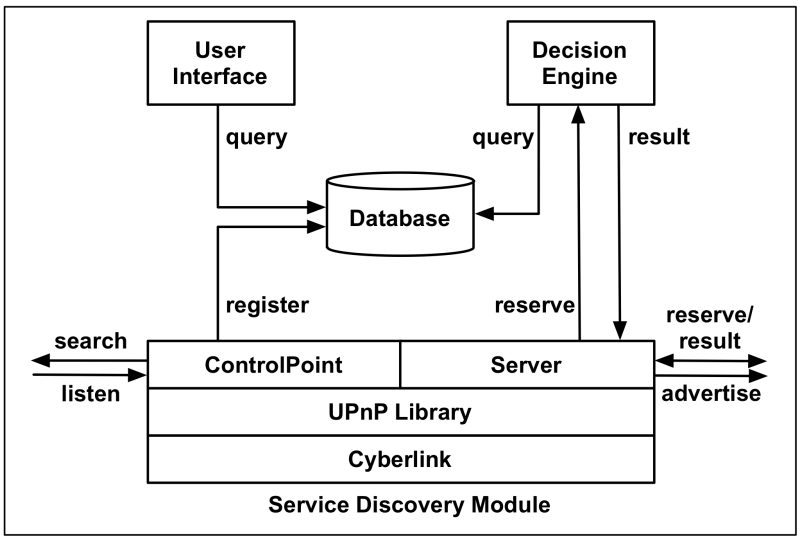


Figure 3.3: *The architecture of the reservation manager implementation based on UPnP.*

manager Service describes methods, arguments and variables to reserve any registered services.

- *Database*. This module contains the reservation information and provides an interface to manage it. The information management includes verifying and storing new service information, querying reservation information, and updating the service states. Each service advertisement being received is related to one service registry that contains the following fields: server device identity (universally unique), service identity (locally unique within the server), service state, service lifetime, reservation time, and service user. The device identity and service identity are used to uniquely distinguish services including the ones from the same server. The service user describes the client identity that reserves the service.

- *Decision Engine*. This module defines the policy for handling reservation request.

- *User Interface*. This module displays the current configuration and enables the remote configurability of the reservation manager server. The configuration may include the deployment of new decision policies, the modification of server leasing times, etc.

3.6 Performance Analysis

3.6.1 Reducing Overhead Traffic

By preventing service access conflicts, unnecessary traffic generation evolving from this conflict can also be prevented, therefore reducing the total average network bandwidth occupation by control traffic. To quantify this, we created a scenario and calculated the amount of traffic created by (conflict) incidents without a reservation manager present in the network, using Equation 3.1. Then we compared it with the same situation but now including a reservation manager, using Equation 3.3.

The scenario is the following. There is one desired server, which is accessed by $n = [2 \dots 10]$ clients that randomly request the server with an average arrival rate of $\lambda = 5$, 10 and 20 clients per hour, respectively. Each client will reserve the server on average for $T_{res} = 2$ minutes, and the server lifetime T_s is set to one hour. Each client will need $T_o = 5$ second to retry to access the server upon interruption. Additionally we assume that each client changes its reservation once during the given duration. We also assume that: data packets for configuring the server, namely CP; for reserving the server, namely RP; and for announcing the possible reservation changes, namely AP, have the same size.

Figure 3.4 depicts the results of the two calculations. The figure shows the number of incidents as a function of the number of clients in the network. The number of incidents of the system is defined as $Traffic/CP$. For calculating Traffic in Equation 3.1, the client random arrival time T_i is assumed to follow a Poisson process, as given in Equation 3.4, with $i = [1 \dots 10]$ is the sequence of the client's arrival. T_s is the interval duration, λ is the arrival rate, and $T_{i=0} = 0$.

$$T_i = T_{i-1} + \frac{(\lambda T_s)^i \cdot e^{-(\lambda T_s)}}{i!} \cdot T_s \tag{3.4}$$

Without reservation manager, the number of incidents increases significantly with the number of clients and increasing client arrival rate. With reservation manager, the number of incidents increases only slightly with client addition (in our case it equals $3n$), and is many orders of magnitude smaller than without reservation management.

3.6.2 Application Overhead

The reservation manager is an active device in the network, which will introduce extra application overheads to the existing system. These mainly consist of extra control traffic generated by the reservation manager advertisements and additional latency for the client application, as each client needs to reserve a service before using it. To asses, we performed some measurements on our UPnP based reservation management system.

We measured the UPnP server advertisement packet for the reservation manager to be 360 bytes. Furthermore, as UPnP recommends, to minimize packet-loss effects, the server should repeat the same advertisement three times. Thus, every advertisement generates 1080 bytes of traffic. If the lease time is set to 30 minutes, then the traffic overhead is fairly small (5 bit/s). However, as the number of clients in the network increases, they will invoke service discovery, and the number of advertisement activities by the reservation manager increases linearly, as shown in Figure 3.4.

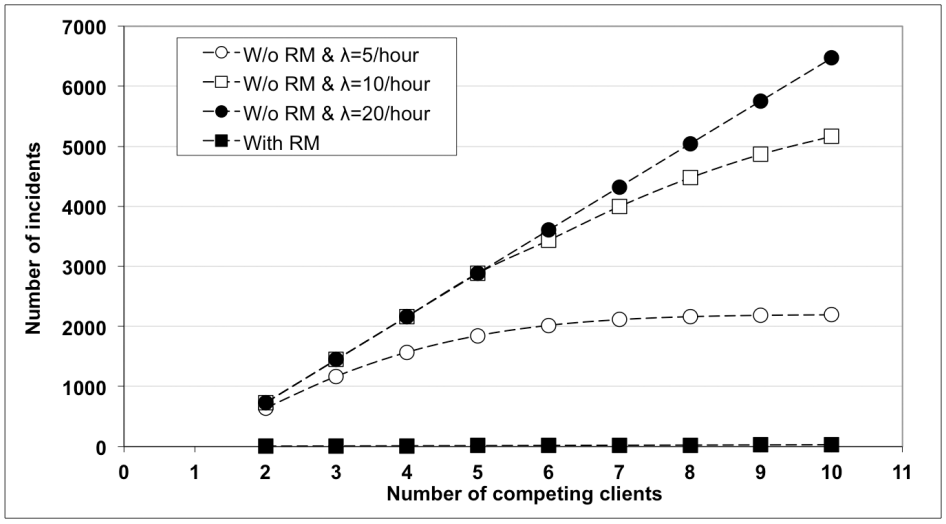


Figure 3.4: *Number of incidents in a system with and without reservation manager upon conflict situation.*

To capture an idea of the client application latency introduced by the reservation manager, we measured the added delay for finding the reservation manager service (namely discovery delay) and for reserving the service (namely reservation delay). We have measured the discovery delay using two UPnP methods namely `Search:All` and `Search:uuid`. With `Search:All`, the client will trigger all available servers to multicast their services randomly within a given maximum time (namely MX time). The `Search:uuid` is a discovery method where the client only triggers the identified server (by a unique identity) to unicast its services within a given MX time. In our implementation, the servers and clients are connected by 100 Mbit/s full duplex Ethernet. We assume up to 100 UPnP devices in the network. For minimizing the sampling error, we measured the discovery delay 100 times for each UPnP device. We set the MX time to 3 seconds for both discovery methods. Figure 3.5(a) shows the average discovery delay using the two methods as a function of the number of devices. Both searching methods give responses within 3 seconds, independent of the number of devices. The average discovery delay of both service discovery methods is 1.6

seconds. This means that the chosen MX time (namely 3 seconds) is sufficient for avoiding scalability problems.

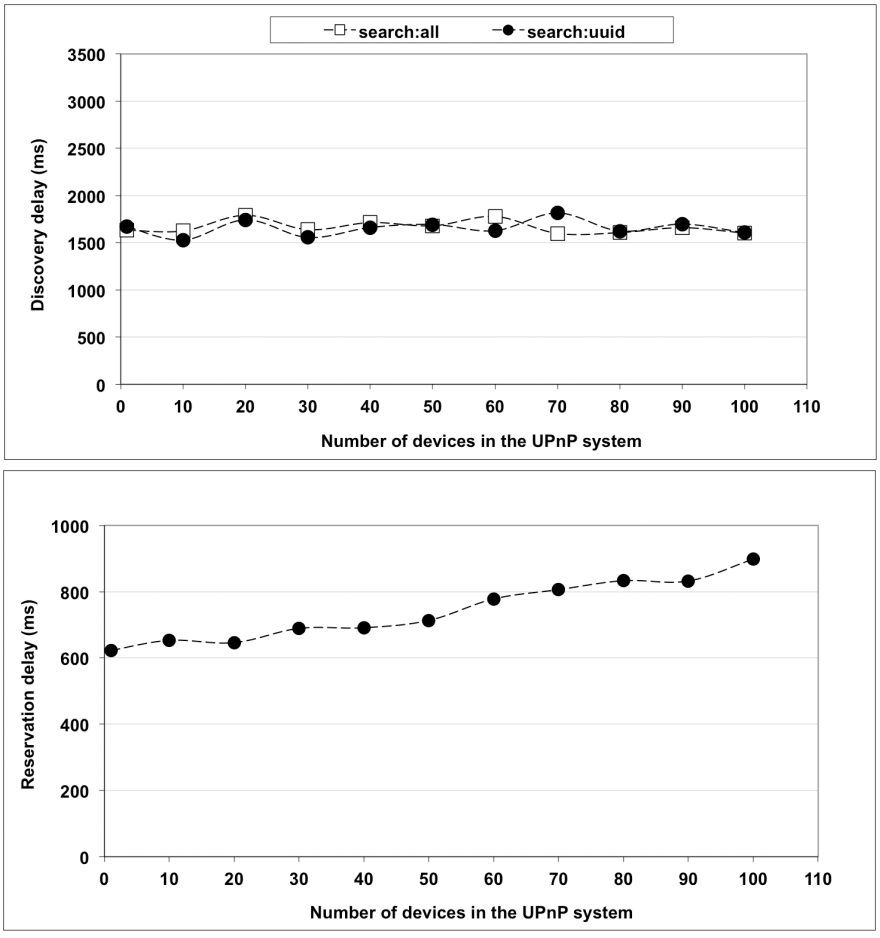


Figure 3.5: *The overheads of the reservation manager: a) search delay and b) reservation delay.*

To assess the reservation delay, we measured the time taken by the reservation manager to respond to the client's reservation request. Figure 3.5(b) shows the reservation delay as a function of the number of UPnP devices. Each measurement was repeated 100 times. In all measurements give about 5% standard deviation. The reservation delay is found to increase slightly with the number of devices in the network. This can be explained as follows. The total reservation delay consists of the network transmission delay (roundtrip) and the application delay introduced by the reservation manager application. From a separate investigation, we found that the transmission delay dominates the total delay and is relatively constant, while the application

delay causes the increment of the total delay. The latter is caused by the fact that every device in the network is represented by a separate Java object in the reservation manager, leading to increasing processing time when adding devices.

3.7 Conclusions

This chapter presents and discusses a remotely manageable resource reservation manager concept for existing service discovery protocols. The reservation concept allows a client to query service availability before using it. Whenever available, the client can reserve the service and later release its reservation. As such, conflicts and contention for service access can be prevented from happening. Also, unnecessary reconfiguration traffic evolving from these conflicts can be avoided.

As the concept can generally be implemented in every SDP, it can potentially manage resource reservation across SDPs. Therefore, it is suitable for heterogeneous systems such as home networks. In that case, home gateway (HG) seems to be suitable for hosting the reservation manager. Also the fact that current gateways already have remote management functionality makes them attractive.

The most innovative part of this work is that the reservation manager benefits from common operations of existing SDPs. Therefore, it does not require any modifications of current standard networked devices and is, thus, backward compatible. However, the reservation manager concept does require clients to cooperate.

From our performance evaluation based on a UPnP platform, it can be concluded that the presence of the reservation manager introduces only a limited amount of extra control traffic on the network and a relatively small reservation application delay. Furthermore we found that the reservation manager reduces the number of conflict incidents drastically. Therefore, it could mitigate the bandwidth occupation significantly, and in turn, improve the system scalability.

Chapter 4

Remote Discovery and Management of End-User Devices in Heterogeneous Home Networks

End-to-end broadband service delivery requires remote management of devices in the home network, behind the home gateway (HG). The service provider can only put limited requirements to these off-the-shelf devices, and therefore has to make intelligent use of their given control and management protocols. In this chapter, we propose architectures for the unique remote discovery and management of such devices in a highly heterogeneous home network. We suggest to have the HG discovering these devices and to proxy management and control actions of the devices to provider's remote servers. This is a substantially different approach from the one taken by the Broadband Forum so far, and has been adopted by the Home Gateway Initiative. Herein, a proof-of-concept is described for the remote management of UPnP devices in the home with a TR-069/UPnP proxy on the HG. However, the architecture is sufficiently generic for any combination of web-services like protocols. The findings of this chapter have been published in Delphinanto et al. (2009b).

4.1 Introduction

The need for new broadband services such as IP TV, IP telephony, gaming and fixed-mobile convergence has lead to a major upgrade of telecommunication networks all

over the world. Within the homes, these new services demand for more and different devices to be supported by a faster private network containing more and more wireless as well as so-called no-new-wires components (such as power line communication). Consequently, home gateways (HG) have become increasingly more intelligent and technically advanced, because they separate the increasingly advanced public network from the increasingly advanced private network. Intelligent broadband modems are often dubbed Residential Gateways or Home Gateways (HGs).

One would expect HGs to become more expensive too. However, current business models and business cases do not allow this to happen. Standardization of HG technology is therefore a crucial development to enlarge the market of specific HG implementations while at the same time keeping costs down. The Home Gateway Initiative (HGI) was founded in 2004, and aims for worldwide, access-network agnostic harmonization of service provider requirements on HGs. The HGI is supported by more than 60 companies. The current set of requirements can be found in HGI (2008).

One of the most essential topics being dealt with in HGI is remote management of the HG and other Customer Premises Equipment (CPE) in the private network. More complexity of HG and home network should not lead to an increase in helpdesk calls or truck rolls. Therefore, one of the workgroups in HGI is dedicated to remote operation and device management.

Many requirements in HGI (2008) are about end-device management, namely remote management of devices "behind" the HG, from a service provider's point of view. These are not requirements for the end devices themselves, but for the HG, and stem from the following considerations. First, service providers want to guarantee end-to-end service quality, not only for services delivered to officially "provider-supported" end devices, but also to devices that the customer bought off the shelf. Second, the home network will remain as technically heterogeneous as it is today, or worse. This does not just concern the various kinds of transmission technologies, but also a manifold of control and management protocols.

The end-device management requirements in HGI (2008) only cover the remote discovery of end devices by service provider. The discovery mechanism that forms the basis of these requirements is described in Section 4.3 of this chapter. The next step in the process, not yet in the HGI releases, is the remote fault-, configuration-, performance-, and security- management of these devices. In Section 4.4, we propose an architecture for this and describe proof of concept implementation. Finally, in Section 4.5, we summarize our conclusions. However, we first give an overview of the state-of-the-art in remote CPE management.

4.2 State-of-the-Art

4.2.1 Remote Management of Home Gateways

From a telecommunications management point of view, the HG is generally the end-point element of the telecommunications network. The network operators do not have decision over in-home devices. The state-of-the-art of the remote management of home gateway is already described in Section 1.2.2. However, the summary is the following. The Simple Network Management Protocol (SNMP) (IETF.org, 2002) is the most popular management protocols of the last decade, but it owns several problems such as the absence of a widely accepted architecture for the delegation of management scripts, local Management Information Bases (MIBs) are vendor specific, and security issues are still open. Web-services based management protocol has become an alternative solution. An example is CPE Wide Area Network (WAN) Management Protocol (CWMP), which is defined in TR-069 of the Broadband Forum (Broadband-forum.org, 2007) and has been developed for the sole purpose of HG management. The CWMP solves the issues that cannot be handled the SNMP. In TR-069, the remote management server is called Auto Configuration Server (ACS).

4.2.2 Remote Management of End-User Devices

Until recently, end device management was mainly local, and based on the use of control protocols such as Dynamic Host Control Protocol (DHCP) and Universal Plug and Play (UPnP) (UPnP.org, 2008). Although control protocols are meant for real-time functions such as connection control, session control, service control, signaling, resource discovery and resource management, they are sometimes also used for typical non-real-time management functions, such as fault management, configuration management and security management (Alanqar & Jukan, 2004). Examples are UPnP Eventing, UPnP Security services, the use of DHCP option 60 and 43 for configuring URLs for initial provisioning, and controlling the UPnP WLAN Configuration Service of a UPnP WLAN Access Point. Other end-device management is often based on vendor-specific web services and a pull model: the end device must initiate the management session. During the last couple of years, however, the Broadband Forum has investigated the use of CWMP for the remote management (by the ACS) of end devices located behind the HG, in the private network. Examples are Voice-over-IP phones (VoIP), set-top boxes, and network-attached storage devices.

Proper end-device management with TR-069 is only possible if the ACS knows which CPE is part of the same private network. For instance, an ACS establishing Quality of Service (QoS) for a voice service needs to provision both the VoIP phone as well as the QoS parameters of the HG through which that device is connected. The most

obvious way to associate the CPE with each other and communicate this to the ACS is by requiring the HG to discover the other end devices and represent the discovered information in the HG data model. Annex F of (Broadband-forum.org, 2006a) therefore requires managed end devices to contain a DHCP client that includes its device identity information in V-I Vendor-Specific Information DHCP Option 125 in DHCP requests. Annex F also defines a way in which the ACS can initiate management sessions with end devices having private IP addresses.

The need for remote discovery of end-user devices in the private network also includes CPE that cannot be managed with CWMP. Discovery of proprietary web-services managed devices cannot be standardized, but discovery of devices that are manageable with DHCP and or UPnP can. Vendors of UPnP devices (mostly consumer electronics) have relatively low margins and are therefore not interested in adding another web services protocol stack like TR-069 just for the sake of remote management. They would rather extend UPnP with more advanced remote management services. It is therefore not to be expected that UPnP devices will be manageable by CWMP in the short term. Discovery and remote management of UPnP devices should thus be tackled by HGI and the UPnP Forum.

4.2.3 HGI Remote End-User Device Management

Figure 4.1 shows the architecture of remote end-device-management that is supported by HGI. The HG is required to communicate with only one ACS, also for multi-provider cases (Balemans et al., 2006). This is shown by arrow 1, representing a management connection between the ACS and the Remote Management Client (RMC) of the HG. By means of a User Interface End Device (UI ED), the user has some control over the configuration of the HG RMC, either directly (arrow 2) or via a web portal to the ACS (arrow 3). The HG is transparent for the remote management of CWMP end devices supported by the HG operator (arrow 4).

The HG contains proxies for the remote management of non-CWMP end devices that support services provided by the ACS operator. The management connections are then represented by arrows 1 and 5. Remote management of CWMP devices by third parties is given respectively by arrow 6 and of non-CWMP devices (e.g. management via web services or SNMP) by arrow 7.

From the HG operator point of view, the managed devices are the ones which are grayed in the figure. If a third party wants to (temporarily) manage these devices, or use the proxy functionality of the HG for remote management purposes, it can do so via the Operation Support Systems of the HG operator (Balemans et al., 2006) under a corresponding Service Level Agreement. Having third party service providers installing their own proxy software on the HG and accessing the HG directly for management requires an open HG service platform and a clear and reliable access-control

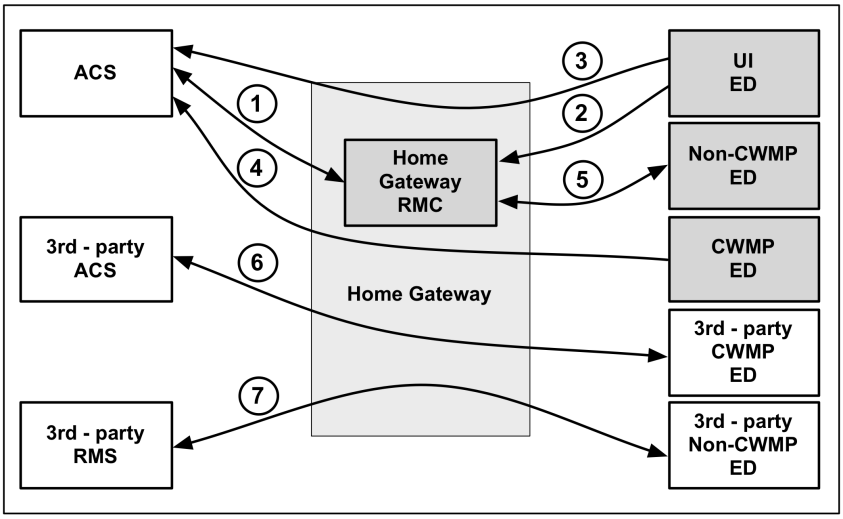


Figure 4.1: *Remote end-device management architecture.*

mechanism. As of HGI (2008) regarded the technology for this as not mature enough for inclusion in the home gateway specification.

4.3 Remote Discovery of End-User Devices

4.3.1 Device Types

Manageable devices are distinguished based on the remote management client(s) they support, so that a discovery mechanism can be specified separately for every type. Seven types are distinguished in HGI (2008).

- Type D is managed by DHCP only.
- Type U is a common UPnP device (which includes a DHCP client by default).
- Type CD is a typical device remotely managed by CWMP, including a DHCP client stack.
- Type CU is a UPnP device that can be remotely managed using CWMP (rare).
- Type C is a device remotely managed by an ACS, but without DHCP client stack (rare).
- Type S is a Session Initiation Protocol (SIP) device, which has User Agent (UA) that can be uniquely discovered.

- Type Z is a user-configured or pre-configured unmanageable IP device, including proxies to non-IP devices.

The end-device discovery architecture is drawn in Figure 4.2. The HG discovers the ID and other information from connected end devices by retrieving and combining information from its Address Resolution Protocol (ARP) cache, DHCP repository, UPnP Control Point cache, and Back-to-Back User Agent (B2BUA) cache. These registries get their information from the various devices connected to the HG.

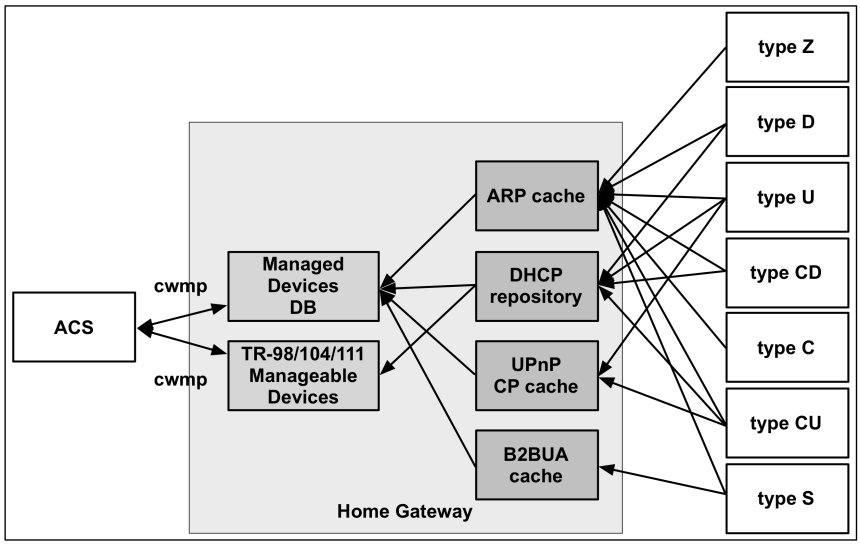


Figure 4.2: *End-device discovery architecture.*

The end-device information discovered by the HG should be made available in the CWMP data model. TR-098 (Broadband-forum.org, 2006b) already contains the `InternetGatewayDevice.ManagementServer.Manageable-Device.{i}` object that specifies the `ManufacturerOUI`, `ProductClass` and `SerialNumber` for every managed device in the home network that contains a CWMP client. HGI proposes the addition of a new object in which all discovered manageable devices are represented, regardless of their management client.

4.3.2 Solution for Device Identity Conflicts

Conflicts may arise because the same information about a device can be discovered by multiple protocols and therefore appear in various registries. We designed a priority scheme in which first priority is given to the information retrieved from the DHCP repository. The reason for it lies in a combination of completeness and reliability. The

DHCP repository is likely to have separate entries for most devices in the network, since many devices contain a DHCP client nowadays. Furthermore, the DHCP repository is likely to contain more information, obtained via various DHCP options, than just hardware address and IP address of a device. Second in priority is the ARP cache, followed by the UPnP cache. The information provided by the B2BUA cache is different anyway.

HGI specifies that the HG must discover and identify devices uniquely. The question then rises as to what determines the device identity. From a management point of view, it does not make sense to identify a device as a physical identity, namely a box. A PC, for instance, can contain many different logical devices (quite often attached accessories such as hard drives, printers and scanners) that support different services and need different management. We therefore define a device as a logical container of capabilities that are interdependent and typically support one end-user service. Many devices, and for sure every UPnP device (root or embedded), can be distinguished on UUID. If the UUID is not available, then the devices must be distinguished on {`ManufacturerOUI`, `SerialNumber`} or, if this is not globally unique, {`ManufacturerOUI`, `ProductClass`, `SerialNumber`}. If also these combinations are not available, then the devices must be distinguished based on hardware address, such as MAC address. For SIP devices, or better, SIP UAs, a globally unique identifier has not been standardized yet. A SIP URI only associates a user name to an IP address. For SIP device identity, Globally Routable User agent URIs (GRUU) are currently considered for standardization in Rosenberg (2009). They are very similar to UUIDs.

After a device has been discovered and uniquely identified, it can be decided if an existing entry in the database should be upgraded or a new entry has to be added. To keep the size of the database under control, and to have the database representing the current status of the home network closely, it should also be decided when to delete a device from the database. In Broadband-forum.org (2006b), a device is deleted from the database when the discovery mechanism provides explicit information about the disappearance of the device, for example, when a UPnP `byebye` message is received from a UPnP device, or when a `DHCPRELEASE` message is received from a DCHP device. When the database overflows, the oldest inactive entry must be deleted. A device is labeled as inactive if it has not been (re)discovered by Gratuitous ARP or DHCP Inform within a time equal to its DHCP lease.

4.4 The Remote Management of UPnP Devices with CWMP

4.4.1 Comparison of UPnP and CWMP

CWMP is a fairly straightforward web-services protocol stack and defines RPCs over SOAP1.1/HTTP1.1/{SSL3.0, TLS1.0}/TCP/IP (SOAP = Simple Object Access Protocol).

Baseline methods concern description, control and eventing. CPE and ACS can `get` information about available methods and parameters with description methods. The ACS can control the CPE with methods that can `get` and `set` parameter values. New instances in the data model can be built and deleted with a method for respectively adding or deleting objects. Other control methods are available for rebooting and file downloading. An eventing method is the information request that the CPE sends to the ACS when it initiates a session. It informs the ACS about the reason for the session and lists, when appropriate, the values of the changed parameters.

UPnP is a client/server based interoperability framework for devices and services in a relatively small-scale best-effort IP sub-network. It distinguishes three logical entities in the network: UPnP Services, which represent the service functionality of a device, UPnP Devices, which act as services servers, and UPnP Control Points (CPs), which act as clients for controlling the services. Here, UPnP Device does not refer to a physical device, but a UPnP server software running on it, providing UPnP Services to UPnP CPs. UPnP defines Simple Service Discovery Protocol (SSDP), SOAP, and General Event Notification Architecture (GENA) for discovery, control, and eventing, respectively. Device and service descriptions are expressed and partially standardized in eXtended Markup Language (XML) templates.

Both UPnP and CWMP use SOAP for control. CWMP uses a single SOAP message for discovery and description. However, in UPnP this is a two-step process, involving SSDP multicast messages for the discovery and SOAP for the description. For eventing, UPnP uses GENA instead of SOAP. There are also some minor differences between the UPnP XML templates and the TR-069 standard data model template (Broadbandforum.org, 2006a), for instance in the allowed data formats. But the most important difference is the fact that UPnP standardizes control actions, and the TR-069 data models deal with management actions. However, there are some exceptions, as already highlighted in Section 4.2.

4.4.2 Proposed Remote Management Architecture

We propose an architecture for remotely managing UPnP devices using TR-069. Although this architecture is protocol specific, it can be easily adapted for other web-services based protocols, and with somewhat more effort also for protocols such as Bluetooth (Delphinanto et al., 2007a).

For remotely managing UPnP devices there are three basic architectures to choose from. The first one that assumes all intelligence is located in the public network and requires no or little changes to the end user devices and the HG. For instance, a UPnP CP is directly connected to the ACS and all UPnP messages are tunneled through the HG. There are many drawbacks. It causes severe overhead traffic in the access network. It also needs a permanent management tunnel to be maintained and an ACS/CP

interface to be defined. Furthermore, it only provides the service provider with the limited management features that UPnP defines today. Last, but not least, the service provider will receive many messages, and therefore the responsibility to act upon them. The overhead traffic can be reduced somewhat by applying intelligent filtering techniques on the HG as currently under standardization in the UPnP Remote Access working group. But that defeats the purpose of this type of architecture.

A second approach is described by Nikolaidis et al. (2007), and assumes the availability of powerful HGs running a complete CWMP/UPnP bridge. It does not need a tunnel to be maintained and reduces the overhead traffic slightly, but does not relieve the service provider of his undesired extra responsibilities, nor it provides him with extra management functionalities. It also introduces new disadvantages, namely the need to have the ACS extended with a "UPnP management operations module" and, because of the heavy requirements on the HG, the inability to apply this architecture to other protocols.

As third approach, we propose an architecture that consists of a much simpler CWMP/UPnP bridge on the HG, and an extension of the UPnP Device with a UPnP Remote Management Service (RMS), to be defined. The bridge only translates the RMS related messages into CWMP. The advantages are clear: the resource requirements on the HG are small; the overhead traffic on the access network is limited to the bare minimum; the provider only receives the messages he is interested in; the end devices can be managed as if they were CWMP devices; and the ACS needs only minimal extensions. The only disadvantage is the need for a UPnP RMS on the end device. However, for the manufacturers this is a much lesser requirement than having to implement a double protocol stack.

Preferably, UPnP RMSs are standardized by the UPnP Forum, but this is not required. The RMSs should be constructed in such a way that the device management information, as discovered by the HG, can easily be transformed into a dynamic, TR-106 compliant (Broadband-forum.org, 2006a), HG data model extension. If done correctly, the ACS can then manage, for instance, UPnP Media Players as if they were TR-069 set-top-boxes; UPnP Media Servers as if they were TR-069 network-attached storages; and UPnP WLAN Access Points as if they were HG internal access points.

4.4.3 Proof-of-Concept

To validate our architecture, we built a proof-of-concept demonstrator containing three Pentium IV PCs running Linux OS. The ACS (PC1, with Dimark ACS software) was connected via the Internet, the HG (PC2), and an Ethernet LAN to a UPnP end device (PC3). The HG contained a Dimark CWMP client and a Philips UPnP CP, with a bidirectional API, as shown in Figure 4.3. The CWMP client and the UPnP CP were both written in C and compiled together into a single application on the HG. The UPnP end

device was running the standard UPnP Basic Device, exposing an RMS. This contains two configurable parameters, mimicking set-top-box functionality. Other parts of the implementation concerned the protocol conversion of UPnP GENA event messages into CWMP SOAP eventing, and the transformation of the two-step UPnP discovery and description process into a single CWMP description. The latter is shown in Figure 4.4. With steps 1-3 the UPnP device announces his presence to the HG using SSDP. The HG then asks the device for its parameters (4-5), turns the received parameters (6-7) into a data model object (8) using the TR-106 template, and communicates it to the ACS (9-10).

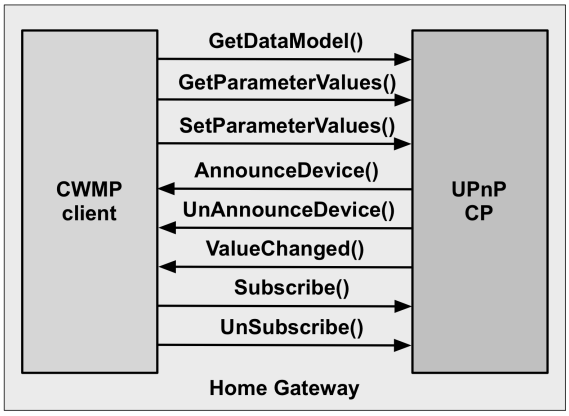


Figure 4.3: *Application Programming Interface (API) between the HG's UPnP CP and CWMP client in our demonstrator.*

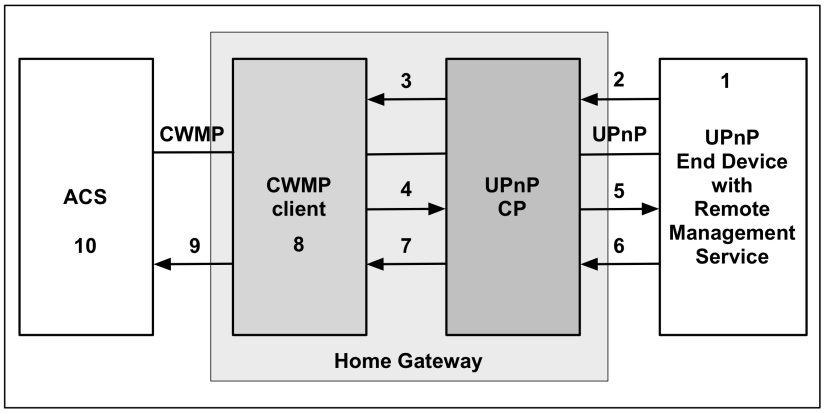


Figure 4.4: *Turning the 2-step UPnP discovery and description process into a single CWMP description.*

When turning on the UPnP device (discovery and description), the new object and its two parameters appear correctly in the web interface of the ACS. At this step, the `InternetGatewayDevice.STBServiceNumberOfEntries` ACS parameter's value changes from `0` to `1`. Changing a parameter value in the ACS (a control method) results in a corresponding change in the UPnP device. Changing a parameter value in the UPnP device (eventing) results in a corresponding change in the ACS. Stopping the UPnP end device removes the corresponding data model object from the HG and sets the `InternetGatewayDevice.STBServiceNumberOfEntries` ACS parameter back to `0`. Thus, the demonstrator is working correctly.

4.5 Conclusions

End-to-end service delivery requires remote management of devices beyond the HG. The service provider can only put limited requirements to these devices. If he would demand, for instance, that the device supports CWMP as a remote management client, then only few people would be able to buy the service. It is more profitable to make use of as many existing devices as possible in the private networks, which support a plethora of control and management protocols. Remote management of those devices requires the HG to discover them and to proxy management and control actions. Of course, this leads to extra complexity and costs of the HG, but in an end-to-end view of the service delivery chain, this is the most optimal approach, and it makes use of the HG in a natural way. This approach also allows service providers to detect problems in the home network (e.g. improper configuration) that may affect the end-to-end service delivery.

We proposed a mechanism for the remote discovery and representation of end-user devices in the home network. The main novelty of the mechanism is that it enables the HG to discover uniquely various types of end devices, using four different discovery techniques concurrently, and to communicate the discovered information to the remote management server in a single data model object. It provides a simple overview of devices in the private network and it prevents contradictory management actions. The mechanism has been adopted by HGI (2008).

We also described an architecture and a proof-of-concept for the remote management of UPnP devices with a TR-069/UPnP proxy on the HG. The main advantage of the chosen architecture is the spread of intelligence over the system. The ACS only needs limited extensions and does not need to function as a control system instead of a management server.

The proxy implementation is very simple but effective. The end devices may need an extra remote management service to be added, but not another protocol suite. Furthermore, this architecture is basically generic for any combination of web-services

like protocols, such as Open Mobile Alliance - Device Management (OMA-DM), Device Profile for Web Services (DPWS) and proprietary implementations. The proof-of-concept works well and will be used to test the qualitative assessment of the scalability of our architecture in a more quantitative way. In this chapter we have used the terms "home network" and "private network" interchangeably. However, the presented management and control concepts can also be applied to other private networks, such as Personal Area Networks (PANs) and in-car networks. Or, more generic, to Personal Networks, which are defined as the collection of interconnected private networks belonging to a single user (den Hartog et al., 2007).

Chapter 5

Path Capacity and Available Bandwidth Estimation for Heterogeneous Home Networks

Current QoS solutions for IP networks are usually based on traffic classification and need to be supported by every device in the end-to-end path to be effective. This is relatively expensive for home networks. Alternative techniques have been proposed that require end-user services to pragmatically adapt their properties to the actual condition of the network. For this, the condition of the network needs to be known in real time. In this chapter, we propose a new method to probe the path capacity and available bandwidth between a server and a client in a home network or any other best-effort small-scale IP networks. The method requires adaptation of the server-side only; is non-intrusive; has a short measurement time; does not require pre-knowledge of the link-layer network topology; and is accurate enough to make educated predictions about the admission of IPTV services and the like. The findings of this chapter have been published in Delphinanto et al. (2010, 2011a,c,d) and patented in Delphinanto et al. (2011b).

5.1 Introduction

Many QoS solutions for IP networks are already available, but seem not to be suitable for home networks. Most of them operate on the principle of traffic classification, where each data packet is placed into a limited number of traffic classes, and each router on the network is configured to differentiate the traffic based on its class. These

solutions need to be supported by every device in the end-to-end path to be effective. This makes them relatively expensive for consumers with many non-depreciated devices: to enjoy QoS they have to buy new devices. Besides, current solutions are different for different layer-2 technologies (wired LAN, WLAN, power-line communications, etc.). Traffic classifiers are defined differently, for instance. Intermediate translators would then be needed to guarantee end-to-end QoS in a heterogeneous path. Though implementations for this exist, they are deemed to be too expensive for mass-scale application today.

Currently, novel cross-layer approaches are under study, which are based on measuring and predicting some intrinsic parameters of the network in real time, and reporting the information to the relevant application, maybe after applying some intelligence. The application can then pragmatically adapt its properties to the actual condition of the network, and report intermittent issues to the user. In Delphinanto et al. (2008), we proposed and analyzed a way to include such functionality in existing service control protocols such as UPnP. These solutions are often easier to apply to current home networks than the ones based on traffic classification, because they only require software updates, and not necessarily on all devices.

A crucial part of such predicting tool is the real-time assessment of end-to-end available bandwidth (throughput) between the relevant client and server in the network. Though many end-to-end speed test applications exist, none of them fulfills all the requirements for use in today's home networks. Amongst these requirements are the following. The tool must:

1. require adaptation/upgrade only of the server-side of the end-to-end path. In a home network that is often the home gateway: services from commercial service providers enter the home network via the home gateway. This makes the tool applicable to the current plethora of existing (thin) clients. For most use cases we can therefore assume the predicting tool to be a service running on the home gateway, serving various clients in the home network, which only need to have a regular IP stack;

2. be non-intrusive. It should not disrupt other traffic in the home noticeably. The other traffic in the home may be important to the user;

3. have a short measurement time, namely it should have a low convergence time from an end-user perspective, and it should be fast enough to react to major changes in the home-network traffic pattern. We assume this to be in the order of a few seconds. This assumption is based on the performance of current Internet speed tests accepted in the market, and the real-time performance of discovery protocols of UPnP;

4. not require pre-knowledge of the link-layer network topology. Home networks

can be very heterogeneous and support many different link-layer topologies, of which some even may not be standardized or widely known;

5. be accurate enough to make educated predictions about the admission of delay- and jitter-critical applications. In the case of IPTV and IP telephony that means an accuracy of $\sim$ 1 Mbit/s and $\sim$ 50 Kbit/s respectively. These figures are derived from the maximum allowed error packet divided by the maximum allowed error duration for SDTV service (Broadband-forum.org, 2006c).

In this chapter, we propose a new tool, "Available Bandwith ESTimator" (*Allbest*), which fulfills all requirements above to the extent that is a software upgrade of only the probe server, it injects less than $\sim$ 1 Mbit/s of probe traffic, produces results with an accuracy better than $\sim$ 1 Mbit/s within less than 10 s, and can be applied on any Ethernet-Wifi combined topology. This the first tool that successfully applies probe round-trip-time (RTT) measurement to wireless LANs. The tool does not assume any home-network topology a *priori*. In the following section (Section 5.2) we first discuss the relevant state-of-the-art, after which the working principle of our method is introduced (Section 5.3). In Section 5.4, we discuss several implementations of *Allbest*. We used the implementations to perform calibration and benchmarking, in Section 5.5. Finally we summarize our finding in Section 5.6.

5.2 State-of-the-Art

The terms capacity and available bandwidth are widely used, however, their definitions mostly remain vague. We follow Prasad et al. (2003) for defining capacity (see Figure 5.1) of a hop as the bit rate, measured at the IP layer, at which the hop can transfer maximum transmission unit (MTU)-sized IP packets (i.e. C_1, C_2, and C_3 for the hops in Figure 5.1 respectively). Therefore, the capacity of an end-to-end path is the maximum IP-layer rate that the path can transfer from source to sink (i.e. C_1 in Figure 5.1). In our work, we assume the MTU size to be 1500 Bytes, of Ethernet v2 (RFC 1191). As a path may consist of several links, the minimum link capacity in the path determines the path capacity. This link is called the narrow link (i.e. hop 2 in Figure 5.1). In contrast, the tight link is the link in the path with the maximum capacity utilization (i.e. hop 3 in Figure 5.1). This is the link with the least available bandwidth due to crossing traffic, namely other traffic in the path considered for admission of a new stream. In many cases, the tight link is in the narrow link, and the link is then referred to as the bottleneck.

Up to 2004 most bandwidth measurement tools were targeted at measuring bottleneck bandwidths in Internet paths. Two types can be distinguished: the ones based on self-loading techniques (also called Probe Rate Model) and others based on packet-

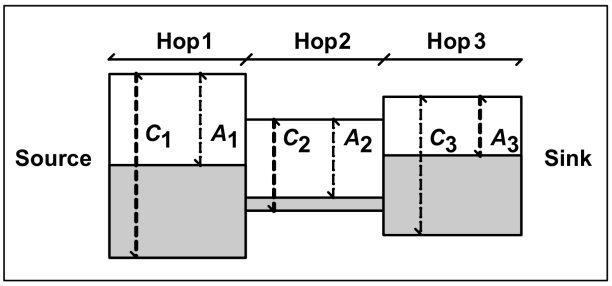


Figure 5.1: *Pipe model with fluid traffic for 3-hop network path.*

pair dispersion (also called Probe Gap Model, or PGM). Self-loading techniques probe the network with trains of packets (Melander et al., 2000) at an increasing rate. The available bandwidth is measured as the rate at which the arrival delays of the pairs within a train start increasing relative to their sending rate. Thus, self-loading techniques rely on a certain amount of flooding the network, and therefore do not fulfill requirement 2. PGM techniques were first explored in Jacobson (1988); Keshav (1991), and send only a few packets, at the rate of the bottleneck capacity (C) or somewhat slower. This allows crossing traffic to get in between the probe packets and disperse them, namely increase the difference in arrival time. If the probe packets are sent at the rate of the bottleneck capacity C, the available bandwidth A can be calculated from the average dispersion rate R at the receiver as Li et al. (2008):

$$A = C \cdot (2 - \frac{C}{R}) \tag{5.1}$$

An issue with PGM techniques is that C needs to be known a priori. From requirement 4 follows that the tool must be able to estimate A without such pre-knowledge of the path. This means C needs to be determined first, and the estimation of A becomes a two-step process.

Various tools also exist for determining bottleneck capacity. Variable-packet-size probing techniques (Bellovin, 1992) calculate the capacity by assuming a linear relation between packet transmission time (delay) and packet size. However, variable-packet-size probing is not suitable for heterogeneous networks (requirement 4), because it cannot deal with layer-2 devices in the path, and assumes that there is no queuing in the system. Network mapping techniques (de Rocha et al., 2007; Wei et al., 2008, and others) estimate the capacity by using various properties of packet-pair inter-arrival times, such as median, entropy, and mean square error, and match them with reference data of the link technologies in the path. Therefore also network mapping needs to know the network topology and the link technologies beforehand. Besides, network mapping needs software implementation at both sides of the path, thus

violating requirement 1.

Fortunately, a variety of the PGM technique called packet-pair dispersion can be used to estimate C, making the estimation of A a two-step algorithm using the same method. This is done by sending two packets back-to-back on the network, thus minimizing the chance that crossing traffic will disperse the packets. It is then the bottleneck that will delay the second packet with respect to the first. The bottleneck capacity C can then simply be calculated from the minimum dispersion D and the packet size L as:

$$C = \frac{L}{D} \tag{5.2}$$

One then should also minimize the chance that crossing traffic disperses the packets further down the path, after being dispersed by the bottleneck. The most obvious way may seem to perform a series of packet-pair probes, and subsequently take the minimum of the dispersions observed. Assuming the cross traffic stochastic, one may then expect a good estimate for C from Equation 5.2. However, this often leads to an overestimation of C, because with some pairs, the first packet will be delayed by cross traffic more than the second packet is delayed by the bottleneck. This is the so-called post narrow-link effect.

Various ways have been proposed to deal with this post-narrow-link effect. *CapProbe* (Kapoor et al., 2004) achieves good estimates for C by selecting the dispersion of the packet pair that displays a minimum in the sum of the delays of both packets. As such, it was also the first capacity-estimation tool based on packet pair probing to claim suitability for heterogeneous paths (requirement 4), because taking this minimum also annihilates the effect of random back-off in wireless links. Besides, to avoid double-sided implementation (requirement 1), the inventors of *CapProbe* suggest to use Internet Control Message Protocol (ICMP) Ping packets and measure the dispersion of the received echo packets at the sender side. However, we learned that for probing a wireless LAN with rate C may yield the correct R when measuring in one direction, but will not if R needs to be derived from round-trip times (it will be explained in more detail in Section 5.3). Unfortunately, the authors of *CapProbe* did not identify this issue and did not describe the roundtrip variety of their tool in more detail.

5.3 *Allbest* Method

5.3.1 Capacity Estimation

Our capacity estimation method (Delphinanto et al., 2010) is based on the packet-pair dispersion technique, namely from Equation 5.2. To allow a single-sided implementation (requirement 1), the roundtrip time measurement as suggested by the authors

of *CapProbe* (Kapoor et al., 2004) is used. We also perform a series of n packet-pair probes to minimize the chance that crossing traffic increases or decreases the bottleneck dispersion of the packets. Assuming the crossing traffic is stochastic, the minimum dispersion D is calculated from the minimum RTT of the first packet (RTT_1) of a probe pair and the minimum RTT of the second packet (RTT_2) of a probe pair. The bottleneck capacity C is then given by:

$$C = \frac{L}{\min_{[i=1...n]}[RTT_2(i)] - \min_{[i=1...n]}[RTT_1(i)]} \tag{5.3}$$

This may look somewhat counter-intuitive, because D may actually never be measured: it may for instance follow from the delay of the 51^{st} packet #1 and the 36^{th} packet #2. One should realize that for a double-sided measurement, dispersions are normally measured instead of delays to avoid having to synchronize clocks at both ends. For an RTT measurement on the other side, synchronization is not needed. One of the immediate consequences of using Equation 5.3 is that our tool converges faster than other methods, because for any packet number j for which:

$$RTT_1(j) = \min_{[i=1...n]}[RTT_1(i)] \tag{5.4}$$

it follows that:

$$RTT_2(j) \geq \min_{[i=1...n]}[RTT_2(i)] \tag{5.5}$$

Thus, the relevant minimum RTTs are observed sooner than the relevant D between two subsequent probe packets.

RTTs can be measured without adaptation of the client side by using MTU-sized Internet Control Message Protocol (ICMP) Ping probe packets, or by sending MTU-sized UDP packets to a non-activated port, like suggested by Kampichler & Goeschka (2003). The client then automatically generates reply packets, respectively ICMP Echo packets or ICMP Error packets (namely code 3 or "Destination port unreachable"). ICMP Error packets are much smaller than ICMP Echo packets and therefore experience hardly any delay on the way back to the probing sender/receiver, assuming that the return one-way capacity between the client and the server $C_{reverse}$ is not much smaller than the one-way capacity $C_{forward}$ between server and client. The final result is then a good measure for $C_{forward}$. This is of particular importance in asymmetric media such as high-speed power-line communication. Another advantage of using UDP packets is that they approach the properties of the stream to be admitted (typically IPTV) the most.

For symmetric media we may also use ICMP Ping probing packets, and assume that the delay and dispersion is the same for both directions of travel. Equation 5.3 then yields $C/2$ rather than C. In case of asymmetric networks, Equation 5.3 yields

C_{tot} when using ICMP Ping probing, with:

$$C_{tot} = \frac{C_{forward} \cdot C_{reverse}}{C_{forward} + C_{reverse}} \tag{5.6}$$

Thus, by using both probing methods we can determine the bottleneck capacities in both directions, even though we require adaptation on the server side only (requirement 1).

5.3.2 Avoiding Contention in Wireless LAN

Equation 5.3 allows us to avoid unwanted contention of probe packets in the wireless medium (Delphinanto et al., 2010). This is the main innovation of our work. From the relevant literature, such as Li et al. (2008); Kapoor et al. (2004), it is not clear why authors have not attempted to do an RTT measurement for a path with a wireless link. We found that existing packet-dispersion techniques will not work in wireless media in round trip, because the reply packet of the first probe packet contends with the second probe packet on the air interface (see Figure 5.2). Irrespective of which packet wins, the reply packet of the second probe packet will eventually arrive at the probing sender/receiver too late. As a result, C will be underestimated.

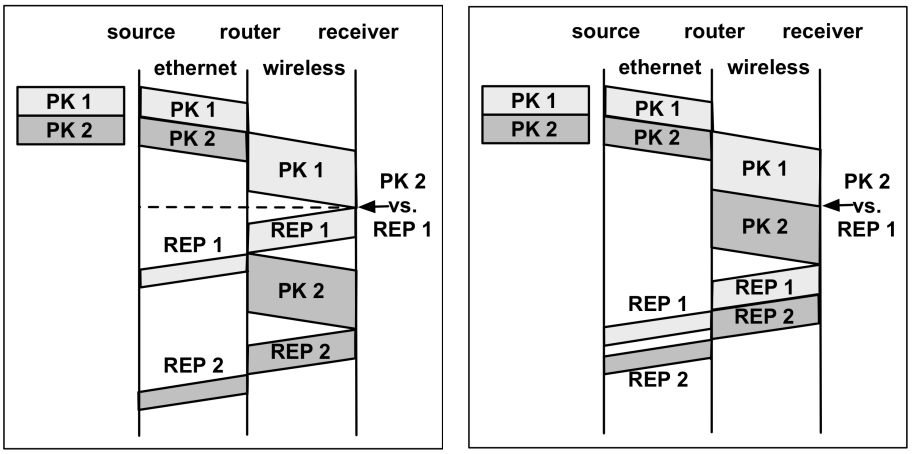


Figure 5.2: *PK1 and PK2 are the probe packets sent back-to-back on the path. REP1 and REP2 are the respective reply packets. (a) PK2 has to wait until REP1 is off the wireless medium and (b) REP2 has to wait until REP1 is off the wireless medium. In both cases REP2 is too late.*

To avoid this contention, we need to prevent the first reply packet from being put on the network. We achieved this (see Figure 5.3) by sending a single packet with size

2·MTU, instead of two packets back-to-back. On the network, this packet will be fragmented (and behave like two individual packets back-to-back), and only after defragmentation a single reply will be sent back at the other end. This will give us the correct RTT_2, namely the RTT_2 that is only delayed by bottleneck dispersion and not by additional contention. Because we are not directly measuring D, but separate RTTs, we can find the correct RTT_1 by sending a series of separate single probe packets (namely not back-to-back) with size MTU.

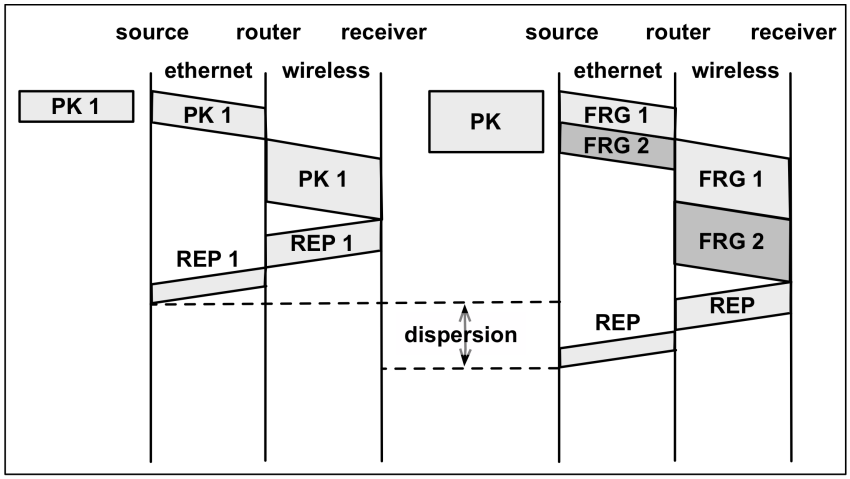


Figure 5.3: *Probe packets PK1 and PK are sent far apart from each other. PK has size 2·MTU and is fragmented on the network. The fragments FRG are dispersed but a reply REP is sent only after defragmentation, and thus no contention has occurred.*

5.3.3 Available Bandwidth Estimation

A deep analysis of the various delays that constitute the RTTs observed during capacity estimation also allowed us to make a good estimation of A (Delphinanto et al., 2011a,c). In Figure 5.4 a typical histogram is shown of RTT_1 that we measured in an IEEE 802.11b network with 1.5 Mbit/s crossing traffic. Besides a clear minimum value, the RTT undergoes two random effects: the random back-off mechanism of IEEE 802.11 (mostly at short additional delays) and the delay caused by queuing due to crossing traffic.

We assume that, for UDP probing, most of the random delay is experienced in the forward direction. This is justified by the fact that the reply packet is very small, and we assume that the queuing mechanism of the system is fair. The reply packet is therefore hindered relatively little by the crossing traffic. We further assume that any systemic delay in the network (for instance processing delay) is either negligible or

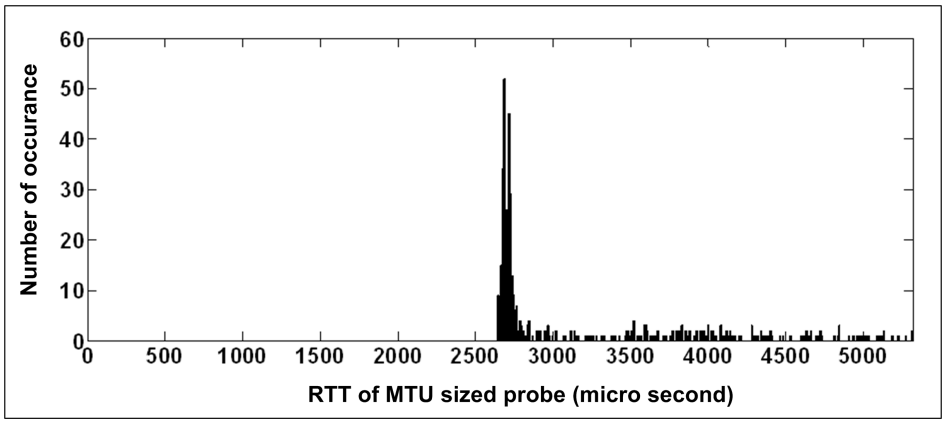


Figure 5.4: *Histogram of 1000 RTTs of MTU-sized probe packets in an IEEE 802.11b network with 1.5 Mbit/s crossing traffic. Bin size = 20 μs.*

canceled when subtracting RTT_1 from RTT_2 (Kampichler & Goeschka, 2003), and that the delay caused by random effects is mainly happening in the bottleneck. A is then given by :

$$A = \frac{L}{\frac{L}{C} + \overline{d_r}} \tag{5.7}$$

with L/C the delay in the bottleneck without crossing traffic following from Equation 5.3, and $\overline{d_r}$ the average delay caused by random effects in the bottleneck. The latter can be derived from RTT_1 as:

$$\overline{d_r} = avg_{[i=1...n]}[RTT_1(i)] - min_{[i=1...n]}[RTT_1(i)] \tag{5.8}$$

Substituting Equation 5.3 and Equation 5.8 in Equation 5.7 , we obtain:

$$A = \frac{L}{min_{[i=1...n]}[RTT_2(i)] + avg_{[i=1...n]}[RTT_1(i)] - 2 \cdot min_{[i=1...n]}[RTT_1(i)]} \tag{5.9}$$

5.4 Implementation

5.4.1 Measurement Test-bed

The set-up of our prototype testbed is schematically drawn in Figure 5.5. Configuration 1 (top) represents a wired IP network and configuration 2 (bottom) represents a wired/wireless heterogeneous IP network.

For configuration 1, we used Ethernet 10BASE-T or HomePlug AV (Linksys Powerline AV Ethernet Adapter) as bottleneck links. The HomePlug AV adapters were plugged into an extension cord. The extension cord was plugged into the office mains

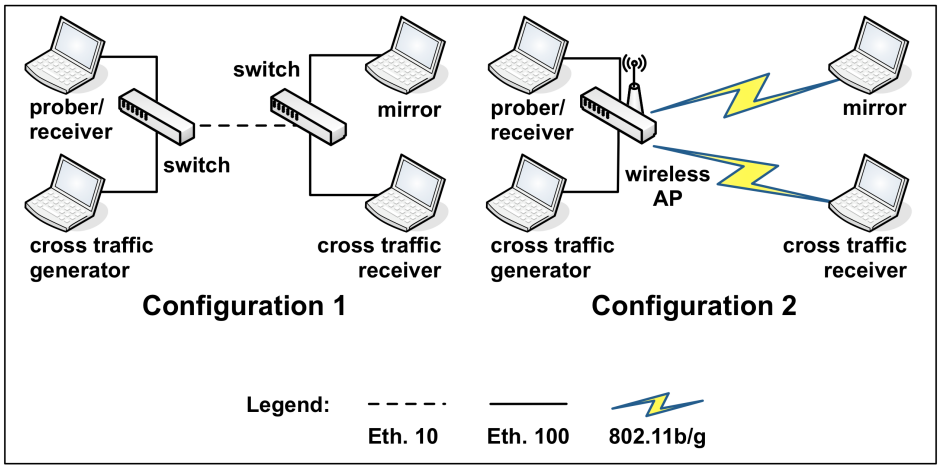


Figure 5.5: *The two configurations for which our tool is tested. The tool runs on the prober and sends packets to the receiver, which subsequently sends reply packets back to the prober.*

via a low-pass filter, with the aim to reduce external noise on the shared medium as much as possible, in order to limit automatic rate adaptation.

For configuration 2, we used WLAN IEEE 802.11b or 802.11g (Linksys WRT54GL v. 1.1) as the bottleneck link. We switched off the automatic rate adaptation and Clear To Send (CTS) protection mode, and run both networks on their maximum physical rates of 11 Mbit/s and 54 Mbit/s respectively. We run the tests in a Faraday cage as well as in our "2.4 GHz band polluted" office environment, but the results did not differ much. IEEE 802.11g is operated in unprotected mode.

In both configurations, our probing implementation, namely *Allbest*, run on the "prober/receiver" computer, and probes the "mirror" via any test-bed configuration. The prober runs an Intel 1.2 GHz dual-core processor and the mirror an Intel 2.0 GHz dual-core processor. As many as possible processes are switched off in both computers, in order to minimize the processing delay and jitter of the probe packets. *Allbest* server consists of a home-built configurable UDP packet generator and a home-built configurable ICMP Ping packet generator, combined with Wireshark (Wireshark.org, 2010) to measure high-precision RTTs. Any $RTT > 2 \cdot min\,[RTT(i)]$ is discarded, and we have verified that most of those long RTTs are caused by uncontrollable processing delay in the laptops due to other tasks of the operating system. A measurement takes 90 probe pairs and is repeated 6 times. This set-up allows us to probe as specified in the previous section, but also to implement a round-trip version of the *CapProbe* tool (Kapoor et al., 2004), which we did.

We benchmarked *Allbest* against the well-known testing tool *Iperf*, *CapProbe*, and against *Wbest* (Li et al., 2008). *Wbest* is the only other real time probing tool we know

which is applicable to wireless networks. It requires the wireless hop to be in the last link, because it needs to be sure that the probing packets arrive at the bottleneck with rate C. For the estimation of C it uses standard PGM and packet-pair dispersion. CapProbre is installed like *Allbest* namely on the prober/receiver while *Wbest* and *Iperf* are installed on both the prober/receiver (which for *Wbest* and *Iperf* just acts as a prober) and the mirror (which for *Wbest* and *Iperf* acts as a receiver). *Allbest*, *CapProbe*, and *Iperf* run on Windows XP service pack 3, and *Wbest* runs on Linux UBUNTU 10.04.

With *Iperf* we measured at which UDP injection rate which packet loss occurs with 1472 Byte payload per packet. The result is fitted linearly and the point where the fitted line crosses the transmission rate axes is interpreted as being the available bandwidth. Each *Iperf* measurement is set for 10 seconds with 1 second interval. The UDP packet loss is averaged over 8-10 similar measurements, leading to 1% standard deviation. Also *Wbest* was configured to use 1472 Byte UDP payload. Each measurement of 90 packet pairs was repeated 30 times. Like *Allbest*, *Wbest* filters and discards unreliable results.

Random UDP crossing traffic is generated with the Distributed Internet Traffic Generator (D-ITG) developed by Botta et al. (2007). The UDP packets have uniformly distributed packet sizes (40-1472 Byte), and are sent at exponentially distributed or Poisson-distributed exponential time intervals. We distinguished crossing traffic and contending traffic, and follow Li et al. (2008) for their definitions, which basically differs the direction relatively to the prober. Crossing traffic shares the bottleneck and propagates with the same direction of probing traffic. Contending traffic also shares the bottleneck but from the opposite direction of the probing traffic. For generating the contending traffic, the crossing traffic generator and the crossing traffic receiver in Figure 5.5 swap their position.

5.4.2 Calibration for the Set-Up for WLAN Test-Bed

For testing the prototype on the WLAN test-bed (configuration 2) with IEEE 802.11b and g, we first measured the actual path capacity of our implementation with the benchmark tool *Iperf*. *Iperf* allows measuring the UDP throughput of a path by reporting the packet loss at a preset UDP injection rate. Varying the UDP injection rate then allows us to find the highest rate at which the packet loss is still negligible. At zero crossing traffic, this value is a good measure for the UDP capacity. The result for this calibration of our set-up is given in Figure 5.6. The UDP capacity of our WLAN b path is 7.0 ± 0.2 Mbit/s. For WLAN g it is 32 ± 3 Mbit/s.

Table 5.1 shows the parameters determining the overhead traffic introduced by the protocols on layers 1 and 2 for the WLAN implementations specific to our set-up. It is important to realize that *Iperf* and any other regular benchmark tool (such as Spirent's Smartbits) measures the effective capacity rather than the raw capacity as defined in

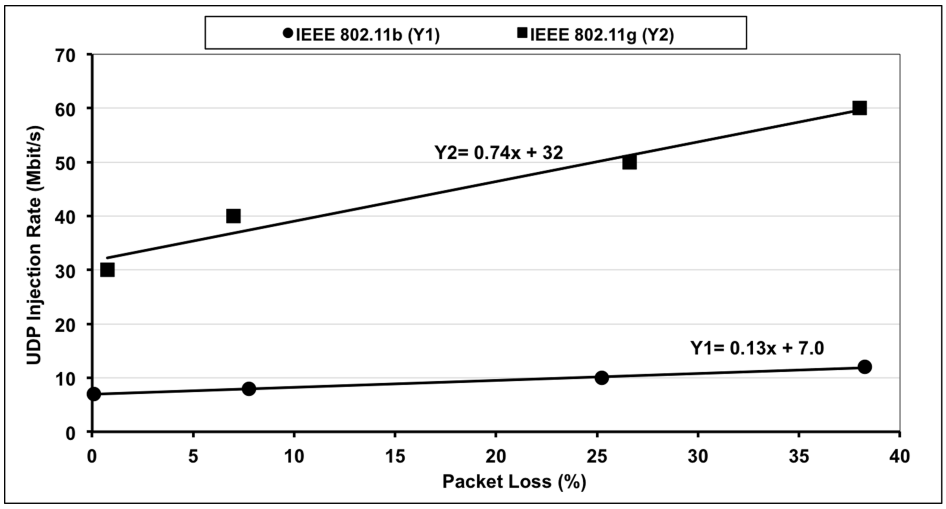


Figure 5.6: *Packet loss as a function of UDP injection rate for an Iperf UDP capacity measurement of configuration 2, for WLAN IEEE 802.11b and g.*

Section 5.2. The difference is caused by the effect of the random back-off mechanism of IEEE 802.11 and many other Ethernet-like protocols for shared media. When the medium is sensed busy, the random back-off mechanism causes a packet to wait a random amount of time slots within a certain contention window until accessing the medium. For a UDP stream of significant length, the net effect is an extra delay of the average random back-off time as given in the one-but-last line in the tables of Table 5.1, leading to the effective capacities as represented. The values measured with *Iperf* are slightly higher than these effective capacity values. We suspect that this is caused by a relative large amount of packets not undergoing random back off, for instance because they are in the beginning of a packet train. Otherwise these results may suggest that the capacity of our implementation is somewhat larger than what we theoretically expected.

For a single packet though, the medium is not sensed as busy, and the random back off does not apply. Because our tool searches for the minimum RTTs of single probe packets, it should yield the raw capacity rather than the effective capacity. From Table 5.1 we expect this to be 7.64 Mbit/s for WLAN IEEE 802.11b and 37.6 Mbit/s for IEEE 802.11g, though the *Iperf* measurements possibly point to slightly larger values.

5.4.3 Demonstrator

We also built a demonstrator, which is a slightly modified version of the set-up drawn in Figure 5.5, and presented it in Delphinanto et al. (2011d). The demonstrator is

Table 5.1: *Parameter values used for IEEE 802.11b/g (Nicolai, 2009)*

	IEEE 802.11b			**IEEE 802.11 g without protection**		
	Data (bits)	**Rate (Mbit/s)**	**Delay (ms)**	**Data (bits)**	**Rate (Mbit/s)**	**Delay (ms)**
DIFS			50			28
SIFS			10			10
PLCP (DATA)	192	1	192	40		20
PLCP (ACK)	192	1	192	40		20
Data	12000	11	1091	12000	54	228
MAC + FCS	272	11	25	272	54	5
ACK	112	11	10	112	54	8
Total			1570			319
Capacity (Data/Delay)		7.64			37.6	
Avg. Random Back-off			310			140
Effective Capacity		6.38			26.1	

shown in Figure 5.7. The *Allbest* server runs on the "*Allbest* Prober" laptop, and it probes the mirror via a heterogeneous topology. A computer running a D-ITG traffic generator (Botta et al., 2007) sends crossing traffic to the mirror. A network attached storage (NAS) is used as a media server for streaming a movie to the client. XBMC (Xbmc.org, 2010) installed on the mirror is the corresponding media player. The NAS server, traffic generator and *Allbest* prober are interconnected with 100 Mbit/s Ethernet. The mirror is connected via the bottleneck link. In the current version of *Allbest*, a measurement can take as many probe pairs as dictated by the desired accuracy, but with a maximum of 10 seconds, and thus is instantly informing the users of the network status.

During the demonstration we first configure IEEE 802.11g (Linksys WRT54GL v. 1.1), and later HomePlug (Linksys Powerline AV) as bottleneck links. We switch off 802.11g's automatic rate adaptation and Clear to Send (CTS) protection mode and run the network on the maximum physical rate of 54 Mbit/s. While *Allbest* is performing the probing, the graphical user interface (GUI) of *Allbest* will show the capacity as well as the available bandwidth of the path from the prober to the client.

The demonstrator is shown from a cold start. The D-ITG generator and the NAS server are turned off, while the prober starts probing the client. The GUI on the prober will show the capacity and the available bandwidth of that path. During the demonstration in Delphinanto et al. (2011d), despite the inevitable 2.4 GHz interference in

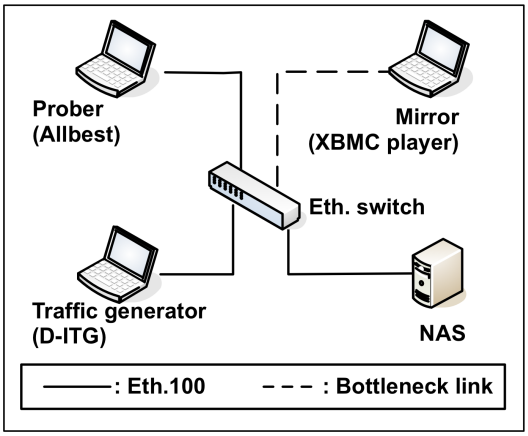


Figure 5.7: *Our Allbest demonstrator. The probe generator runs on the Allbest Prober laptop. The crossing traffic generator is a netbook running D-ITG. Path capacity and available bandwidth estimation are shown on the GUI of the prober, while a video is being streamed from the NAS server to the XBMC media player on the client.*

the demonstration environment, the estimated path capacity was around 38 Mbit/s, which is the expected throughput of IEEE 802.11g on the network layer.

The available bandwidth will be whatever the interference in a demonstration room dictates. It will most probably be enough to stream a 4 Mbit/s MPEG-4 movie from the NAS server to the client, which we therefore start. As the streaming goes on, the *Allbest* GUI will display the (unchanged) path capacity and the new, reduced available bandwidth. With D-ITG we then inject crossing traffic to the path being probed. We adjust the amount of crossing traffic until it is depleting the bandwidth resource of the path. This will be indicated by an estimated value of the available bandwidth close to what is minimally needed for the video stream not experiencing packet loss, and obvious degradation of the video quality on the media player.

The whole procedure is then repeated with HomePlug. The bottleneck link is created by two HomePlug adapters plugged into a multiple-socket extension cord. The extension cord is connected to the mains power socket via a low-pass filter, to keep external noise out of our system.

5.5 Performance Analysis

5.5.1 Capacity Measurement without Crossing Traffic

Figure 5.8 shows the results for the bottleneck capacity as measured with our tool *Allbest* for configuration 2, as a function of the number of probe pairs per measure-

ment i, with IEEE 802.11b (diamonds) and IEEE 802.11g (squares) as bottleneck links, without crossing traffic. The triangles are the results for IEEE 802.11b obtained with our implementation of *CapProbe*. The dashed lines are not fits through the data, but represent the theoretical values for the capacity, as given by Table 5.1. A measurement of 500 probes takes in total about 7 seconds for IEEE 802.11b and about 4 seconds for IEEE 802.11g.

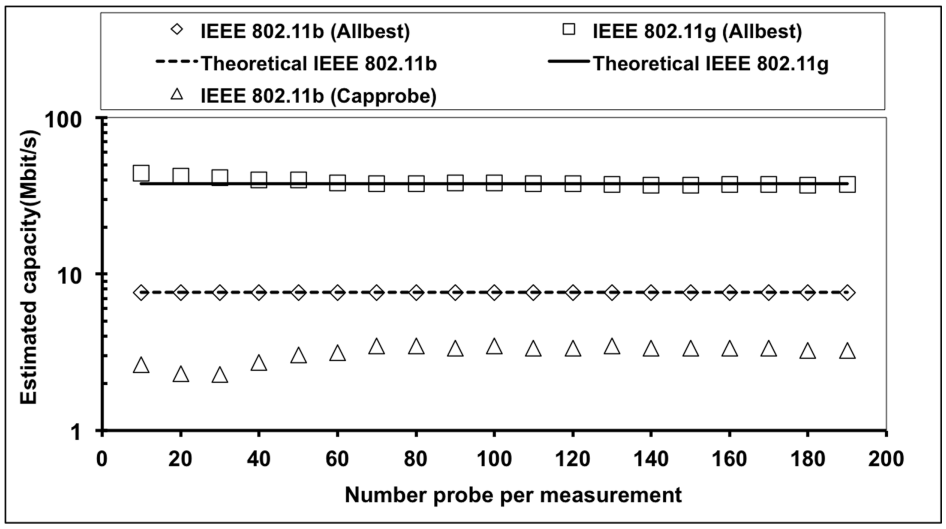


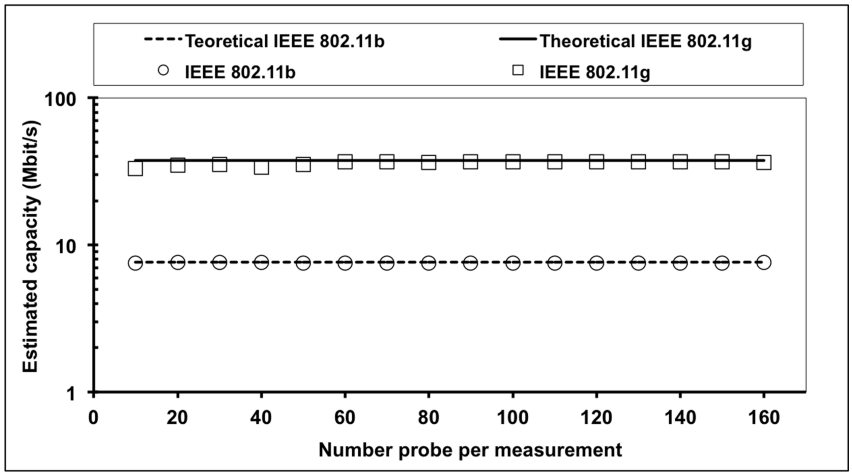
Figure 5.8: *Bottleneck capacity as measured in our prototype Allbest for configuration 2, as a function of the number of probe pairs per measurement with IEEE 802.11b (diamonds) and IEEE 802.11g (squares) as bottleneck links, and with zero crossing traffic. The dashed and dash-dotted lines represent the respective theoretical capacities given in Table 5.1. The triangles are the results for IEEE 802.11b obtained with our implementation of CapProbe.*

For IEEE 802.11b, our implementation of *CapProbe* converges in about 100 probes (13 seconds) to a $C = 3.4 \pm 0.2$ Mbit/s. This value is clearly too low and can be explained by the additional delay that the second reply packet is suffering due to contention as shown in Figure 5.2. *Allbest*, however, converges significantly faster (within 10 probes, less than 1 seconds) to a more accurate value of 7.6 ± 0.1 Mbit/s. For IEEE 802.11g, *Allbest* results in $C = 38 \pm 3$ Mbit/s (within 60 probes, again less than 1 seconds). These results fit perfectly with the expected theoretical values, and are better than we have observed.

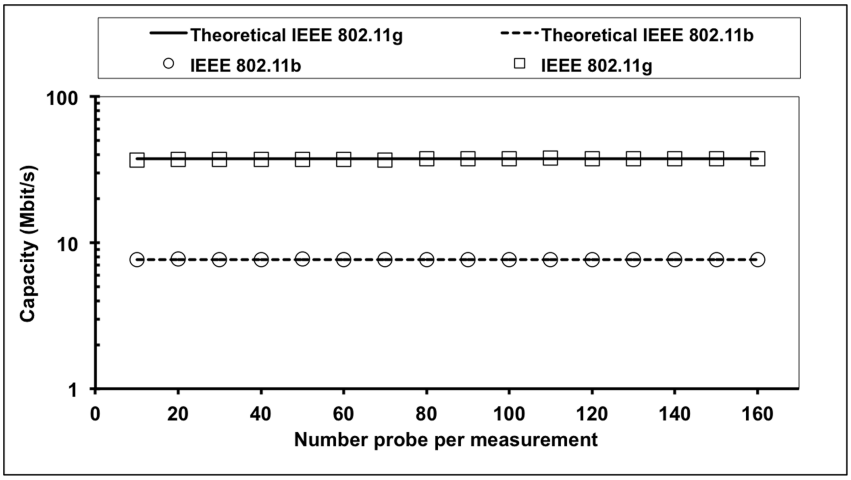
5.5.2 Capacity Measurement with Crossing Traffic

We then injected 50% crossing traffic in the same testbed used in Section 5.5.1. For the configuration with IEEE 802.11b this amounts to an average crossing traffic rate

of 3 Mbit/s and for IEEE 802.11g it is about 11 Mbit/s. Because of the high rate and the stochastic characteristic of the crossing traffic, the router's buffer sometimes overflowed and we lost 10% of the probes. Figure 5.9 shows the observed bottleneck capacity as a function of the number of pairs needed, with the inter-arrival time of the crossing traffic exponentially distributed (a) and Pareto distributed (b).



(a)



(b)

Figure 5.9: *Bottleneck capacity as measured in our prototype Allbest for configuration 2, as a function of the number of probe pairs per measurement, with IEEE 802.11b and IEEE 802.11g as bottleneck links and 50% crossing traffic with (a) exponentially distributed and (b) Pareto distributed inter-arrival intervals. The dashed and dash-dotted lines represent the respective theoretical capacities as given in Table 5.1.*

For IEEE 802.11b, the estimated capacity converges to 7.6 ± 0.1 Mbit/s within 20

samples and 10 samples for exponentially and Pareto distributed inter-arrival times of the crossing traffic, respectively. Within the error margin this is equal to the theoretical capacity. For IEEE 802.11g, *Allbest* converges to $C = 37 \pm 2$, for which it needs 60 respectively 10 samples, again equal to the theoretical capacity within the error margin. In both cases the convergence time is less than 1 seconds, although it is slower for exponentially than for Pareto distributed inter-arrival times of the crossing traffic. Exponentially distributed inter-arrival times are relatively short, and therefore the chance is smaller that indeed two packets 1 and 2 will be found that did not experience delay from crossing traffic.

The observed standard deviation in our results is accurate enough to make decisions about the admission of IPTV-like streams, but not for IP telephony (requirement 5). The observed convergence time fulfills requirement 3. For existing PGM tools, for these 1-2 seconds we still only know the bottleneck capacity. To determine the available bandwidth, another probing experiment is needed, which may easily take up to 5 seconds. In traditional Ethernet/Wifi heterogeneous home networks, changes in path capacity are much less likely to happen than changes in available bandwidth. It may therefore be sufficient to do the capacity probing only occasionally. However, modern home-networking technologies such as broadband power line communications often exhibit fast rate adaptation. This would yield for continuous capacity as well as available bandwidth probing. This is another important innovation of *Allbest* that it can estimate the available bandwidth from the capacity probing measurements directly.

5.5.3 Available Bandwidth Measurement

We have obtained the available bandwidth for three different topologies in configuration 2 with *Iperf*, *Wbest* as well as *Allbest* (in its UDP probing variety):

1. Prober/receiver → 100BASE-TX → IEEE 802.11b → mirror
2. Prober/receiver → 100BASE-TX → IEEE 802.11g → mirror
3. Prober/receiver → IEEE802.11g → 100BASE-TX → mirror

For every topology we generated three different amounts of crossing traffic (X), at about 0%, 25% and 50% of the capacity, and 25% of contending traffic. Figure 5.10, 5.11 and 5.12 summarized the results for topology 1, topology 2, and topology 3, respectively.

For topology 1, with IEEE 802.11b, all tools yield similar results at first sight and within the error margins. *Allbest* estimates the capacity C of 802.11b on 7.6 ± 0.2 Mbit/s, which is equal to the theoretical value (see Table 5.1). For all tools, the available bandwidth A is lower than C for $X = 0$. This is caused by the random back-off

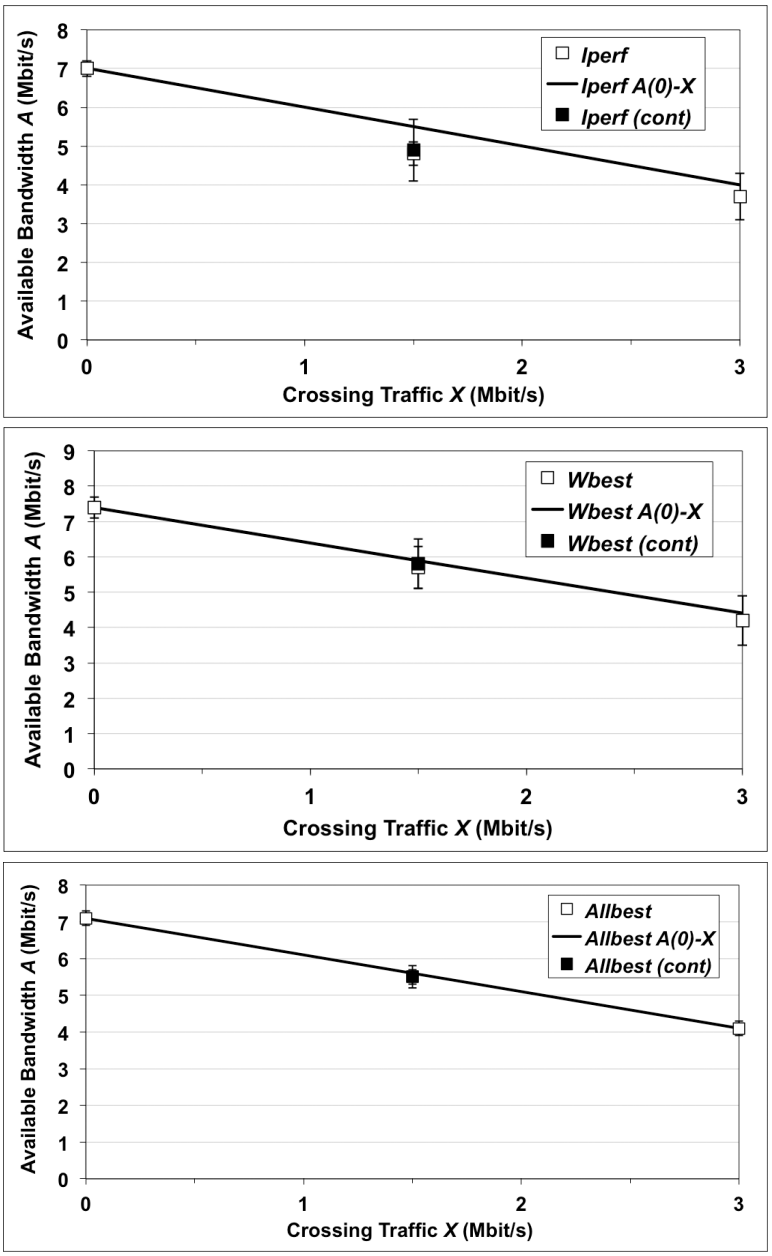


Figure 5.10: *Available Bandwidth measured with different tools and different amounts of crossing and contending (CT) traffic for topology 1.*

mechanism of WLAN. If the random back-off algorithm were to be active for all packets, we would expect $A = 6.4$ Mbit/s. Since all tools estimate available bandwidths somewhat higher than that, we suspect that there are still many packets that do not undergo random back-off.

On the whole, *Allbest* seems to find larger available bandwidths than *Iperf*, and *Wbest* finds even larger ones. Even though *Iperf* is a well-known benchmarking tool, it is probably underestimating A in our experiments. Because the crossing traffic is stochastic, some packet loss will already be recorded at relatively low *Iperf* injection rates. For both *Allbest* and *Wbest*, the values for A at $X > 0$ are also closer than *Iperf* to the expected value of A (obtained by simply subtracting the crossing traffic rate X from the available bandwidths A at $X = 0$ Mbit/s). The error margins of the *Allbest* results are remarkably lower than the ones for *Wbest*, though we tried to have the results based on the same number of probes. We do not have an explanation for this yet.

For *Allbest*, the value for A at $X = 3$ Mbit/s was calculated by discarding any $RTT > 3 \cdot min\,[RTT(i)]$, rather than using the default cut-off time of $2 \cdot min\,[RTT(i)]$, as stated in the previous section. We found that with the latter, too many packets had been discarded that were clearly delayed by crossing traffic, and A was grossly overestimated (5.1 ± 0.2 Mbit/s). Many PGM techniques use a default cut-off time of $2 \cdot min\,[RTT(i)]$. This follows from their assumption of fair queuing congestion management in the router. This means that bottleneck delays can never be larger than $2L/C$, even if the utilization by crossing traffic is larger than 50% (which then just results in larger packet loss). Crossing traffic of 3 Mbit/s with randomly distributed time intervals will utilize the bottleneck more than 50% for at least part of the time. Fortunately, the actual congestion management mechanism of the router (most probably store and forward) still allowed us to capture relevant packets with larger RTTs and compute a realistic value for A.

For topology 2, with IEEE 802.11g, *Allbest* is showing a clear supremacy. *Allbest* estimates the capacity C of 802.11g on 38 ± 2 Mbit/s, which is equal to the theoretical value. The value of A at $X = 0$ Mbit/s is then expected to be 26 Mbit/s if the random back-off algorithm is active for all packets. *Iperf* and *Allbest* estimate somewhat higher again, but *Wbest* significantly underestimates A, also for larger X. The inventors of *Wbest* warn for underestimation when the probe packets arrive at the bottleneck at a rate larger than C (Li et al., 2008). Surprisingly, *Wbest*'s capacity estimation for topology 2 is quite good, namely 38 Mbit/s.

The fact that *Wbest* arrived at plausible answers for A with topology 1 can be explained by it grossly overestimating the C of topology 1 (8.8 Mbit/s). The results for *Allbest* are very close to the ones for *Iperf*, and have the lowest error margins of all. But like with Figure 5.10, it is not sure whether *Iperf* yields the correct values. More than *Iperf*, *Allbest* yields values for A at $X > 0$ close to what one obtains by subtracting X from $A(X = 0)$. However, simply subtracting the crossing traffic rate from the available

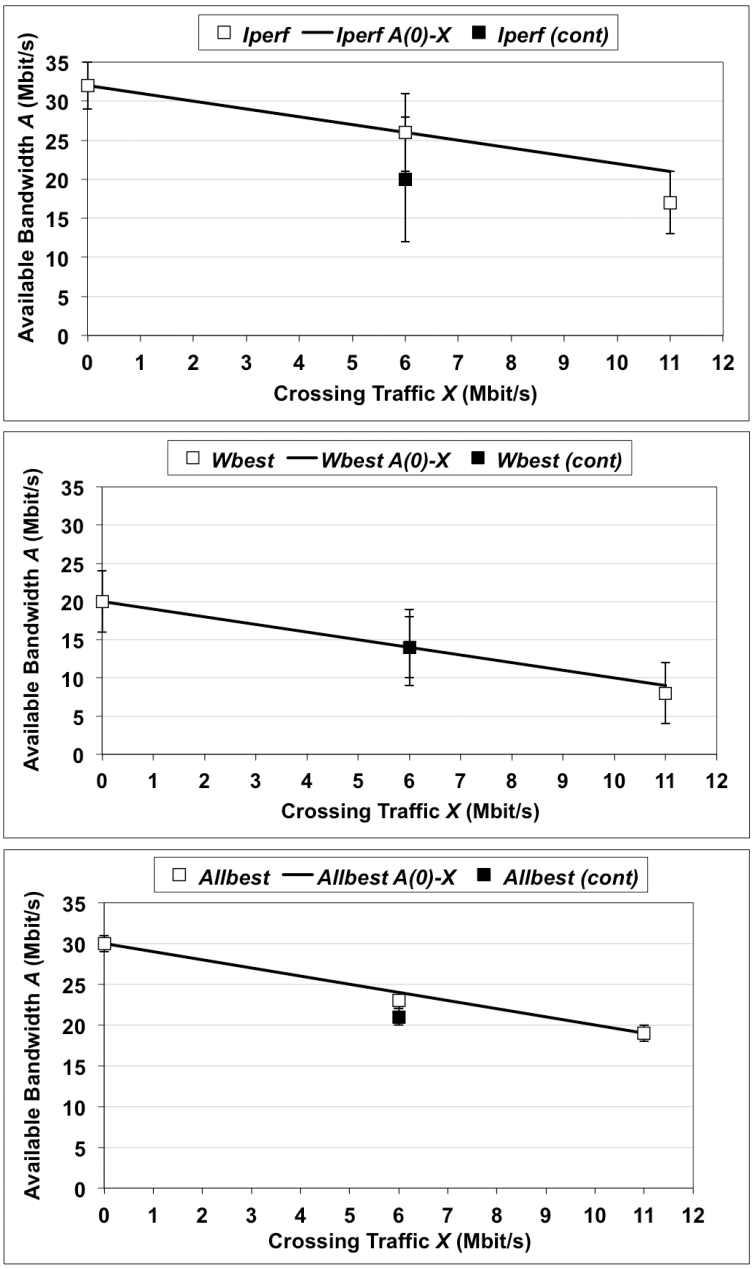


Figure 5.11: *Available Bandwidth measured with different tools and different amounts of crossing and contending (CT) traffic for topology 2.*

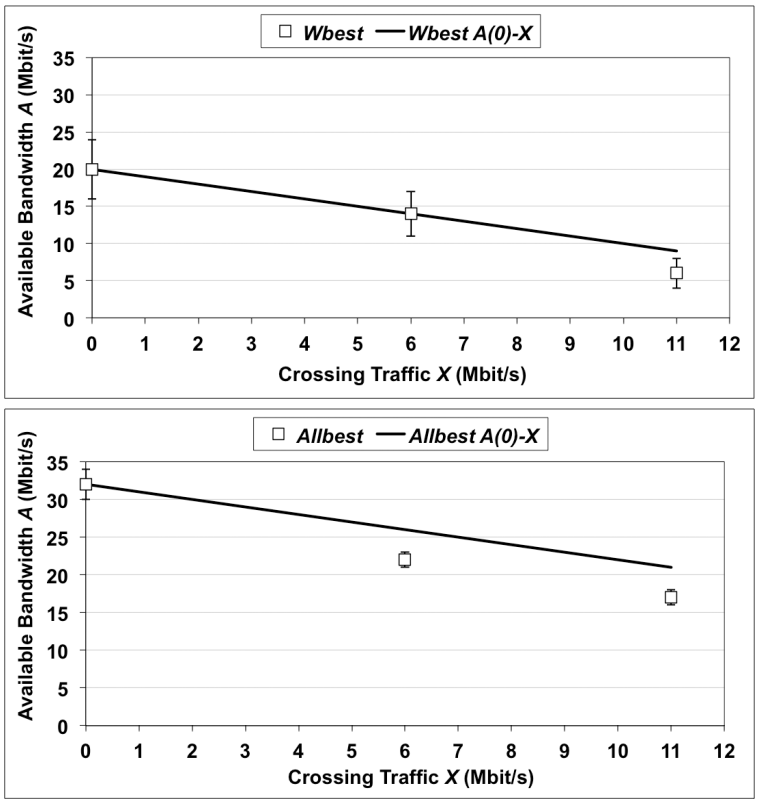


Figure 5.12: *Available Bandwidth measured with different tools and different amounts of crossing and contending (CT) traffic for topology 3.*

bandwidths at 0 Mbit/s crossing traffic seems not to work for any of the tools anymore.

The results for topology 3 are much the same as for topology 2. This shows that *Allbest* can function in either order of physical- and link-layer technologies. Unfortunately we could not get any UDP results from *Iperf*, because it cannot inject faster than about 10 Mbit/s when directly connected to an 802.11g network.

5.5.4 Overhead

Our results are largely obtained by using the ICMP Ping probe variety of the *Allbest* tool, but have been verified with the UDP/ICMP Error implementation. We also applied it to fixed Ethernet bottlenecks (configuration 1, 10/100 Mbit/s), where we observed the same results as with regular speed test methods. The amount of overhead that our tool creates equals to a maximum of 3·MTU = 36000 bits per probe pair in any direction. For 60 probe pairs per 2 second this amounts to a maximum probing

traffic of 1.1 Mbit/s. Although this is certainly not negligible in networks containing relatively slow links such as IEEE 802.11b, the tool is still less intrusive than flooding techniques. However, if the available bandwidth is less than 1.5 Mbit/s, one may expect the *Allbest* tool to become intrusive also. A limit of 1.5 Mbit/s is just good enough to make decisions about the admission of one extra IPTV stream.

5.5.5 HomePlug AV

HomePlug AV has a theoretical throughput on the physical layer of 200 Mbit/s. Its UDP throughput was tested with a Spirent Smartbits Traffic Generator/Analyzer and found to be 76 ± 1 Mbit/s. This is well comparable with the TCP throughputs between 60 and 80 Mbit/s as observed in Murty et al. (2008). HomePlug is a technology that exhibits fast asymmetric rate adaptation. We tried to limit this effect by carrying out our measurements on an extension cord plugged in the mains via a low-pass filter. However, some channel asymmetry will be present, either by interference or by HomePlug AV's Medium Access Control (MAC) scheme, which seems not to be fair compared to, for instance, Wifi (Balachandran et al., 2002). We therefore assume the observed C to be $C_{forward}$. With our UDP/ICMP Error probing tool we found $C_{forward} = 72 \pm 2$ Mbit/s (without crossing traffic), which is very close to the value measured with Smartbits. With ICMP Ping probing we found $C_{tot} = 17 \pm 1$ Mbit/s. Using Equation 5.6 this would yield $C_{reverse} = 22 \pm 2$ Mbit/s. Cross-checking these results by measuring in the other direction did not make much sense because of the unpredictability of the medium and the behavior of the MAC-layer protocol.

5.6 Conclusions

Available bandwidth probing and path capacity estimation are already mature technologies for the Internet. However, probing in home networks puts different requirements to a tool. None of the existing techniques fulfills those requirements. *CapProbe* is the most advanced in that respect, so we used it as a benchmark, next to *Iperf*.

Our *Allbest* path capacity and available bandwidth estimation tool is based on the lightly intrusive packet-pair dispersion technique with five additional innovations. First, it uses round-trip-time measurements. This fulfills the requirements that it can be used with existing client devices in the home and supports asymmetric media. Second, the bottleneck capacity is estimated by separate determination of the minimum RTT of packet 1 and the minimum RTT of packet 2. This helps fulfilling the requirements of fast convergence and acceptable accuracy. Third, the probe packets are not sent back-to-back, but as separated pairs of MTU- and 2·MTU-sized packets. This avoids contention between probe- and reply packets, and makes the tool applicable

to any shared-medium link-layer technology that uses back-off mechanisms on the MAC layer for collision avoidance. Thus the tool requires hardly any pre-knowledge of the home-network link-layer topology. Fourth, it has two varieties based on ICMP Ping/Echo and UDP/ICMP Error respectively, which in concatenation are able to measure forward and reverse path capacities in asymmetric networks. Finally, *Allbest* is the first PGM tool that can estimate the capacity and available bandwidth with the same probing packets. This besides saves the measurement time but also reduces the traffic being generated.

We have built a prototype, and our performance measurements indicate that the tool converges reasonably fast (within seconds). The capacity measurement results approach the expected raw-data capacity of IEEE 802.11b, IEEE 802.11g, and HomePlug AV very well, and are significantly better than the ones obtained by existing tools. For the available bandwidth results, *Allbest* works well and outperforms *Iperf* and *Wbest* for various topologies based on 100BASE-TX, IEEE 802.11b, and IEEE 802.11g, for up to 50% crossing traffic. These results are accurate enough to make decisions about the admission of IPTV-like application-traffic streams.

Chapter 6

Analysis of Topology Discovery Protocols for Home Networks

Service providers are demanding the development of novel diagnostic tools with which they can remotely troubleshoot the home network. One of such tools should be able to gather information about the topology of the home network. In this chapter, we propose a set of key performance indicators for home network topology discovery architectures, and how they should be measured. We apply them to the Link-Layer Topology Discovery (LLTD) protocol and the Link-Layer Discovery Protocol (LLDP), and show that these protocols do not fulfill all the requirements as formulated by the service providers. The findings of this chapter have been published in Castellanos et al. (2012).

6.1 Introduction

The increasing popularity of Internet-Protocol (IP) based services has accelerated the evolution of consumer's home networks. The home network has become a complex environment providing connectivity to several devices with different capabilities. Any given home may now have one or more PCs, laptops, smartphones, tablets, Internet radio devices, set-top-boxes, network-attached storage devices, digital photo frames, game consoles, Internet phones, and printer servers. These digital devices are interconnected via a combination of Ethernet switches, Wifi access points, power line bridges, and cordless telephony base stations, and connected to the Internet and the managed operator network via one or more Home Gateways (HGs). Furthermore, in the near future it is expected that this heterogeneity of devices and networks in the home will prevail with smart meters, energy management devices and e-health devices using network technologies such as Zigbee, Z-wave, and Bluetooth.

For service providers, this is proving to be an increasing service management nightmare. Home networks are owned, installed, configured, and controlled by often ill-educated end users. These networks become a part of end-to-end delivery chain of the services provided by the providers. The network path between e.g. the HG and the relevant end device is beyond the control of the service provider. However, if there is a problem in these networks, it will lead to end users calling the service providers' help desks. Although service providers posses tools to manage their own core and access networks, they lack the means to gather information related to home network characteristics. For them, the home network is largely a black box. Service providers are therefore demanding the development of novel diagnostic tools with which they can remotely troubleshoot the home network. The worldwide Home Gateway Initiative is currently drafting a home-network diagnostics requirement document (Thorne & Bitzer, 2010) spelling out the service providers' needs. One of the required tools should be able to gather information about the topology of the home network. Topology information includes a list of active devices and their capabilities, a list of connections between devices, and on a per-connection base the link technology that is used. In the remainder of this chapter, any device in the home that forwards, switches or routes traffic is called a Home Network Infrastructure Device (HNID). They are distinguished from the end devices on which the services are consumed, here also called STAtions (STAs), and the HG.

Current topology discovery mechanisms have been mainly developed for large-scale homogeneous Ethernet networks, such as business networks. For small-scale heterogeneous networks, very few architectures are available. The most well-known is Microsoft's proprietary Link Layer Topology Discovery (LLTD) protocol, which is included in Windows 7^{TM} operating system. An alternative protocol is provided by the open standard IEEE 802.1AB "Link Layer Discovery Protocol" (LLDP). However, there is no literature available on how well these protocols perform, let alone how they compare. Worse, it has not been established how the performance of these protocols should be measured. In this chapter, we propose a set of key performance indicators for home-network topology discovery architectures, and recommend how they should be measured (Section 6.3). Further, we describe our testbed in Section 6.4 and show our analysis of LLTD and LLDP in Section 6.5. Finally, the results are benchmarked against the service providers' requirements in Section 6.6.

6.2 Topology Discovery Protocols

6.2.1 AFT and STP

The first step in topology discovery, namely device discovery, is well researched and many architectures exist, also for use in home networks (Delphinanto et al., 2009b). A

popular example is Universal Plug and Play (UPnP). The second step, connection discovery, is slightly more complicated. The most straightforward approach may seem to read out the Address Forwarding Tables (AFTs) stored in the link-layer switches in the network, and using the Spanning Tree Protocol (STP) for further topology discovery. These mechanisms have been well researched and described in Lowekamp et al. (2001); Peng et al. (2010), and subsequent papers. The advantage of this approach is that it uses standard technologies that are already supported by every switch in the market. However, this approach only works with Ethernet networks and not for other networks such as Wifi and Power Line Communication (PLC). In addition, for reading out the switches remotely, a remote management protocol is needed such as the Simple Network Management Protocol (SNMP). However, how this protocol should be used has not been standardized and many switches do not support SNMP.

6.2.2 LLDP

IEEE 802.1AB (LLDP) is a layer-2 protocol specifically designed for topology discovery in local area networks. It relies on LLDP agents being supported by the end devices and the HNIDs. The LLDP agents are the protocol end points. They encapsulate and transmit (or decapsulate after reception) LLDP Data Units (LLDPDUs) in a Medium Access Control (MAC) frame. The MAC frame is broadcast on the broadcast domain that the transmitter is in. The LLDPDU consists of Type- Length-Value (TLV) fields containing information such as chassis ID, port ID, port description, system name, system description, system capabilities, and management address. With the received information, an LLDP agent updates a local Management Information Base (MIB), which can then be remotely accessed using a protocol such as SNMP.

A typical implementation has LLDP agents in transmit-only mode running on end devices. HNIDs then must support a full LLDP agent, a MIB, and an SNMP server (for which the HNID needs to be IP addressable). To read out the MIBs, an SNMP client may run in the service provider's network, but to reduce overhead and complexity it should preferably run on the HG. Because of the rich information contained in de LLDPDUs, inferring the topology is easier than in the case of using AFTs and STP, and can also be done for heterogeneous networks, identifying the link types. Another advantage of LLDP stems from the fact that LLDPDUs are broadcast regularly, and MIBs are updated accordingly. The obtained topology map can therefore be assumed to be a good representation of the actual topology of the home network. The main disadvantage of LLDP is the heaviness of the requirements it puts on HNIDs. The HNIDs play a central role in the topology discovery to succeed, but many HNIDs currently in the market do not support SNMP nor LLDP.

6.2.3 LLTD

LLTD (Microsoft.com, 2010) is a layer-2 protocol developed by Microsoft as part of the Windows Rally set of technologies. Its operation is based on a central entity, known as the "mapper", performing a series of tests on demand of the user. The mapper typically runs on PCs. But service providers are interested to have it implemented on the HG too. The other protocol end point is called a "responder". Its task is to respond to received LLTD test queries with the device and link information requested. Like LLDP, LLTD information is encapsulated in MAC frames. In contrast to LLDP, they are not broadcast autonomously, but as part of a test session initiated by the user. It is not publicly known how the mapper infers the topology from the received responses, but we assume that the algorithms used are similar to the ones described in Black et al. (2004).

In LLTD, node and link discovery is a two-step process, first involving the mapper performing a Quick Discovery (QD) and then a sequence of Topology Discovery Tests (TDTs). With QD the responders in the home network are discovered via an exchange of broadcast "discover" frames and "hello" responses. The discovered responders are then interrogated by the TDT in a unicast fashion, using various sequences of "emit", "query" and "reset" frames, on which is responded with "probe", "train", "ack", "queryresp" or "flat" frames.

The main advantage of LLTD is that the requirements on the HNIDs are low. They should preferably run the lean responder stack. But even if they do not support LLTD, there is a significant chance that they will be detected indirectly by the mapper's intelligent discovery algorithm. This also means that the additional use of a remote management protocol is not needed. The mapper produces the topology map directly from the information it received.

6.2.4 HTIP

The International Telecommunication Union (ITU) G.phnt working group is currently working on a new protocol known as Home-network Topology Identifying Protocol (HTIP). A first draft is published in TTCS (2010). Instead of using LLDP between the HNID and the end devices, HTIP uses a modified version of LLDP between the HNID and the HG. Between the HNIDs and the end devices it just uses the AFT information. The end devices themselves are discovered with UPnP. Now, the only requirement on the HNID is to run the modified LLDP agent. How the HG infers the topology from the obtained information has not been described yet. Because of the preliminary status of this new standard, we have not yet studied it any further.

6.3 Performance Indicators

The topology discovery problem can be divided into two problems: 1) discovery and classification of HNIDs and end devices within a home network according to their behavior and supported link layer technologies, and 2) the creation of a graph representing how these devices are interconnected. There is consensus among HGI members that a home-network topology discovery diagnostics tool for service providers should fulfill the following requirements:

1. The accuracy must be close to 100%, namely the obtained map must contain a negligible amount of mistakes.
2. The time between requesting a topology map and obtaining it must be less than 2 seconds.
3. The overhead traffic that the topology discovery procedure creates and the memory resources it confiscates must not disturb other services in the home.

Inspired by these requirements, we defined five performance indicators for home-network topology discovery: 1) accuracy of device classification, 2) accuracy of network graph (or map), 3) discovery time, 4) traffic overhead, and 5) memory requirement.

6.3.1 Accuracy of Device Classification

Let the HNIDs, for example, be an Ethernet switch (namely an Ethernet- Ethernet bridge, here abbreviated as SW), a Wifi access point (namely an Ethernet-Wifi bridge, here abbreviated as AP) or a HomePlug node (namely an Ethernet-PowerLine bridge, here abbreviated as HP). If we then classify all end devices as STAs, that leaves us with four possible devices to discover. Receiver Operating Characteristics (ROC) is a method used to analyze classification systems. Our classifiers try to relate an unknown device to one of the four possible types according to its behavior or advertised information. The result of the match could be positive (P) or negative (N). After comparing the actual type of the device and the identified type of device, we have the following possible outcomes:

- True Positive (TP): An obtained positive match is correct.
- False Positive (FP): An obtained positive match is incorrect.
- True Negative (TN): An obtained negative match is correct.
- False Negative (FN): An obtained negative match is incorrect.

The classification accuracy Acc_{class} can then be expressed as

$$Acc_{class} = \frac{\#TP + \#TN}{\#P + \#N} \tag{6.1}$$

The quality of a classifier can also be represented as a coordinate in a ROC graph, which shows the true positives rates (TPR) on the Y-axis and the false positive rates (FPR) on the X-axis, with

$$FPR = \frac{FP}{\#N} \; and \; TPR = \frac{TP}{\#P} \tag{6.2}$$

A classifier A is equally good or better than B if its position in the ROC graph is closer to (0, 1).

6.3.2 Accuracy of Network Graph

A network can be modeled by using undirected graphs. Networked devices and connections are represented by nodes and links. A graph has a mathematical representation known as adjacency matrix. The matrix has size M x M (M is the number of nodes). Its elements are 1 where a link exists between nodes and 0 where there is no link. We can compare the real topology with the generated map by comparing their adjacency matrices, and find the number of positions where the values in both matrices are 1 ($\#TP$) or 0 ($\#TN$). The graph accuracy Acc_{graph} then equals:

$$Acc_{graph} = \frac{\#TP + \#TN}{M^2} \tag{6.3}$$

6.3.3 Discovery Time

LLTD broadcasts a Reset frame when the LLTD discovery process is initiated and also when the LLTD discovery process ends. The difference between the timestamps of these frames (e.g. measured with Wireshark) gives us the discovery time for LLTD. The Network Management System (NMS), containing the SNMP client for topology discovery with LLDP, sends SNMP queries to all active HNIDs. To estimate the LLDP discovery time, we read the timestamps from the first and last SNMP queries, and take the difference.

6.3.4 Traffic Overhead

The traffic generated by the topology discovery processes consists of sequences of probing and advertisement messages exchanged during the discovery time. It does not contain large media streams, flooding experiments, etc. We therefore measure this overhead traffic as a rate averaged over a relatively long period of time. For the

latter we do not choose the discovery time, because a long process injecting a lot of traffic would, in reality, be more disturbing than a short process injecting little traffic. We therefore decided to use a fixed duration of 60 second, and assume that any discovery process started at $t = 0$ second would be over by then.

6.3.5 Memory Use

The total memory resources required by each protocol are given by the memory space needed by the application that is idle in the background, the memory space required to perform the tests, and the memory space required to store the topology data. The first two are measured with the `Mem Usage` and the `Mem Delta` parameter of Windows Task Manager, respectively. The memory space needed to store the topology data is calculated from the protocol specs. The end result for memory usage is the sum of `Mem Usage`, `Mem Delta`, and the storage space.

6.4 Testbed Implementation

6.4.1 Home Gateway (HG)

The HG in our testbed must include the protocols of interest. HG manufacturers do not yet offer an appropriate HG supporting LLTD and LLDP. We therefore constructed an HG by taking a CISCO SF-300-08 Ethernet Switch for small business with LLDP agent, MIB and SNMP server (3 in Figure 6.1), and connected it with a Linksys WRT54GL gateway, containing a router and a DHCP server (1 in Figure 6.1). Also connected is a Dell Latitude 2100 Netbook (2 in Figure 6.1) running an LLTD mapper in its Microsoft Windows Vista OS and running the Solarwinds Engineer's NMS, which includes an SNMP client and a topology inference algorithm, and Wireshark.

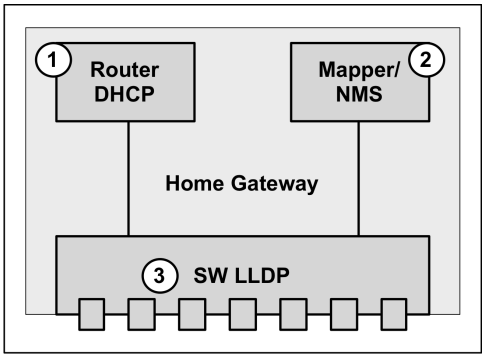


Figure 6.1: *The test bed HG containing 3 different physical components.*

6.4.2 Active Devices and Configuration

Our end devices are mimicked by Acer Aspire ONE Netbooks running Windows XP on which we installed Microsoft's LLTD responder. There are many open-source LLDP clients. We chose the haneWIN LLDP agent because it runs on Windows and supports both transmission and reception modes.

Our choice of HNIDs is strongly depending on what the market has to offer. This makes that our measurements will give a good indication of what service providers may observe in real life today. The disadvantage is that not all theoretically possible configurations can be tested. The HNIDs we used are the CISCO SF-300-08 Ethernet Switch, the Hewlett Packard HP V-M200 802.11n Wireless Access Point, and the Sitecom LN - 513 HomePlug adapters. Table 6.1 shows for all devices used in our testbed their support for LLTD and LLDP.

Table 6.1: *LLDP and LLTD support of devices in test bed.*

		LLDP Agent		**LLTD**	
Device	**Type**	**Tx Mode**	**Rx Mode**	**Responder**	**Mapper**
HG	HG	no	yes	no	yes
Station	End-Device	yes	no	yes	no
Access Point	HNID	yes	no	no	no
Switch	HNID	yes	yes	no	no
Home Plug	HNID	no	no	no	no

6.4.3 Testbed configuration

The basic testbed configurations we used for testing LLTD and LLDP are shown in Figure 6.2. They are based on research that TNO recently did in selection of households in The Netherlands (Castellanos, 2010). The result was a snapshot of home-network topologies used by the "early majority" of technology adopters. It turned out that the vast majority of home networks was close to one of these basic configurations, or was constructed as a simple linear combination of some of these configurations. The number of stations connected varied between 1 and 3 for every configuration.

6.5 Measurement Results

An example of a topology map generated by LLTD for the PLC configuration with 2 stations is shown in Figure 6.3. The upper three devices and their interconnections correctly represent the HG (and its disconnect from the Internet). Also the two stations and their links to the home network are correct. However, the HomePlug power

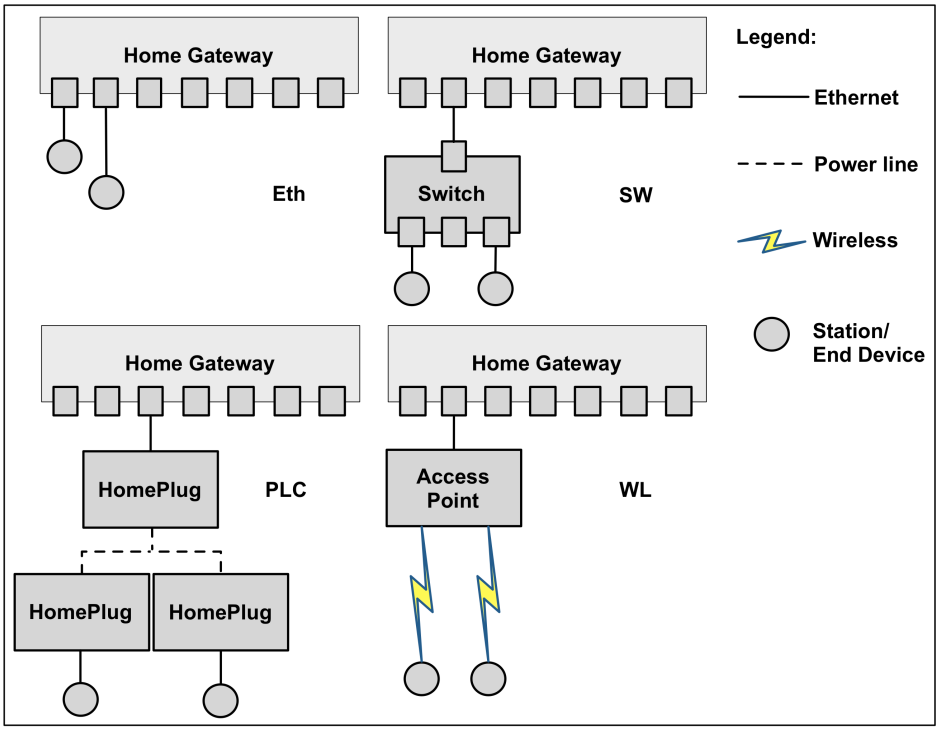


Figure 6.2: *Basic test bed configurations.*

line nodes are shown as an Ethernet hub and two switches interconnected with Ethernet. We consider this graph is incorrect because HomePlug and Ethernet are technically different. HomePlug has a dynamic link-layer rate adaptation while Ethernet does not. Maps generated by LLDP and the SNMP NMS look similar. Similarly we generated maps for every configuration, for 1-3 end devices, and for LLDP and LLTD, namely 24 in total. Every measurement is repeated 5 times to check reproducibility and precision. The latter is found to be $\sim 5\%$ for all results.

6.5.1 The Accuracy of Classification and Network Graph

For every map we found the false and true positives and negatives and the adjacency matrix, and calculated the accuracy indicators according to Equation 6.1 and Equation 6.3. The results are shown in Table 6.2. The ROC graph is shown in Figure 6.4. Note that in Figure 6.4, LLDP gives the perfect score. From the results we conclude that LLDP has a better classification accuracy than LLTD, but a worse network graph accuracy. This can be explained by the relatively superb advertisement mechanism of

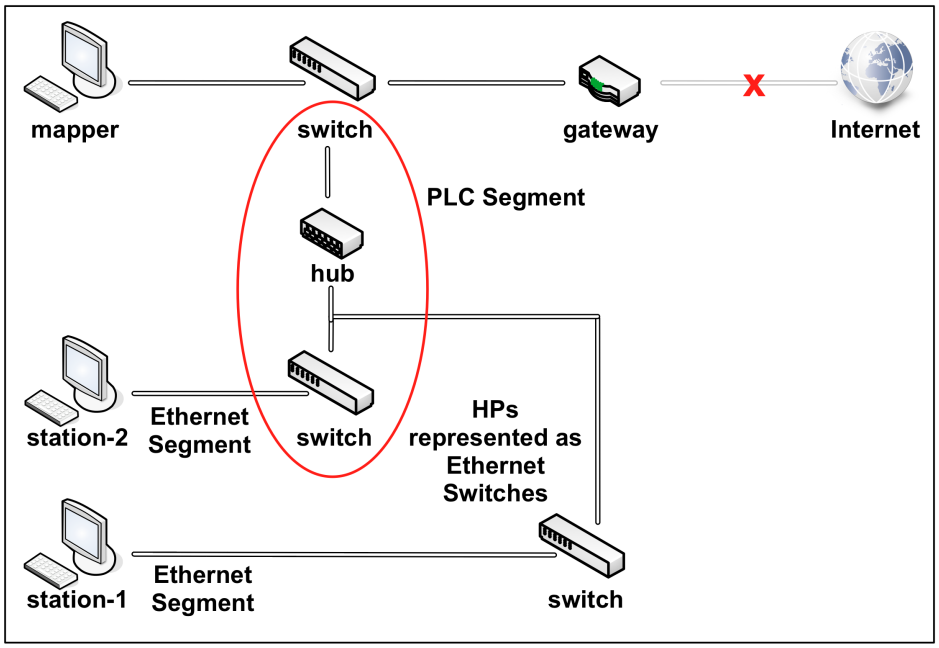


Figure 6.3: *Example of a generated topology map (LLTD, config. PLC).*

LLDP devices, as well as the limited support of LLDP by commercial HNIDs combined with the clever algorithms performed by the LLTD mapper. In none of the cases is the accuracy near 100%, and therefore requirement 1 is not fulfilled.

6.5.2 Discovery Time

Figure 6.5 shows the results for the discovery time. The results for LLTD are different for the different configurations, but are always in the range of 4-10 second. For configurations Eth, SW, and WL the topology discovery time increases linearly, but very little (from 1% to 5%) with every new station added. For PLC, the increment of topology discovery time is more pronounced compared to other configurations (10% to 30%). This can be explained by the algorithm deciding on doing many more tests than for the other configurations, because it estimates that this configuration is relatively unusual and needs extra tests to indirectly infer the existence of the HomePlug nodes. For LLDP, the curves for the configurations Eth, SW and PLC virtually overlap. The discovery times are also much longer than for LLTD, namely in the range of 16-55 seconds. It turns out that most of the time is needed for reading out the MIBs. For WL, the discovery time seems to be independent of the number of active stations. For LLTD we

Table 6.2: *Classification and graph accuracy for LLTD and LLDP.*

	LLTD	**LLDP**
$\#TP$	3	3
$\#TN$	20	15
$\#FP$	1	0
$\#FN$	4	2
$\#P$	4	3
$\#N$	24	17
Acc_{class}	82%	90%

		Acc_{graph}	
Configuration	$\#STAs$	**LLTD**	**LLDP**
Eth	1	100%	100%
	2	100%	100%
	3	100%	100%
SW	1	100%	100%
	2	100%	100%
	3	100%	100%
PLC	1	63%	50%
	2	72%	56%
	3	83%	59%
WL	1	100%	78%
	2	100%	75%
	3	100%	76%

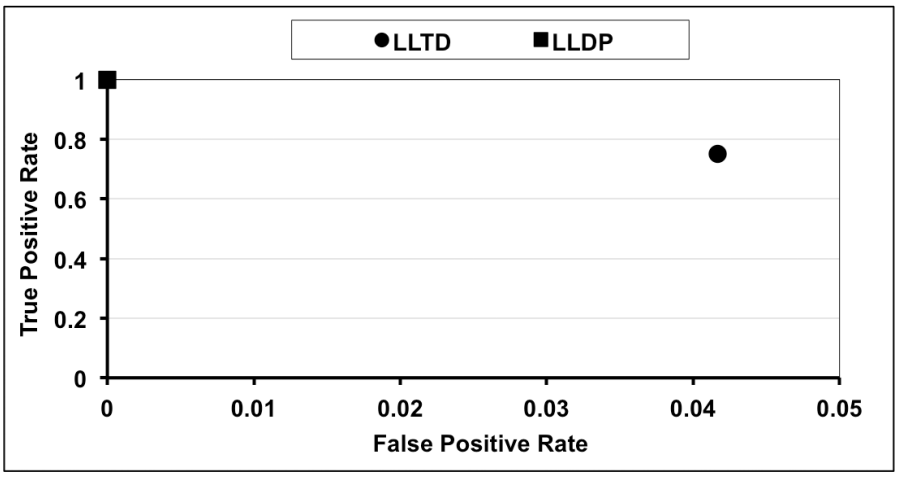


Figure 6.4: *ROC graph for LLTD and LLDP.*

found out that for the WL configuration, the QD test is already enough to discover the topology. The AP only supports the Tx mode of LLDP, and therefore the end devices are not discovered at all with LLDP. Neither LLTD nor LLDP fulfills requirement 2.

6.5.3 Traffic Overhead

Figure 6.6 shows the average injected traffic rate for LLDP and LLTD. The explanation of the results follows largely the same logic as for the discovery time. For any con-

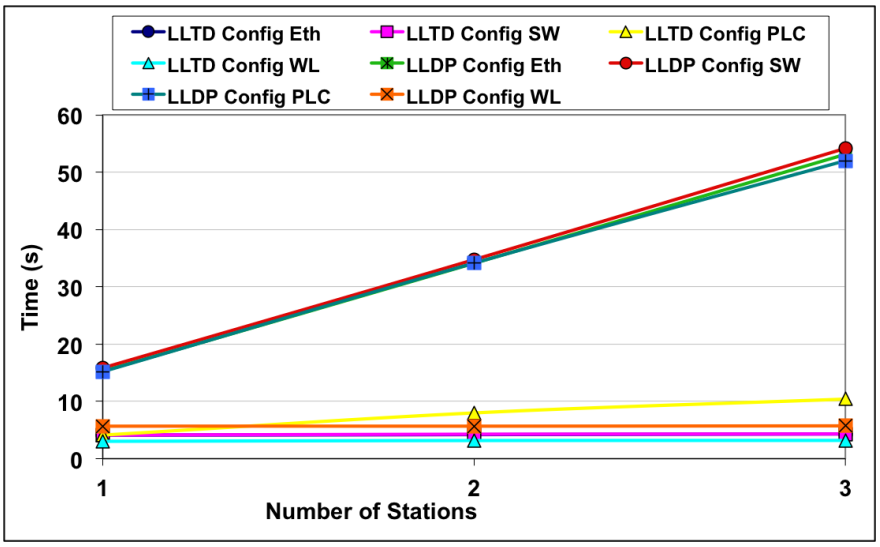


Figure 6.5: *Discovery time for LLTD and LLDP.*

figuration, the absolute size of the injected traffic rate is very low and easily fulfills requirement 3.

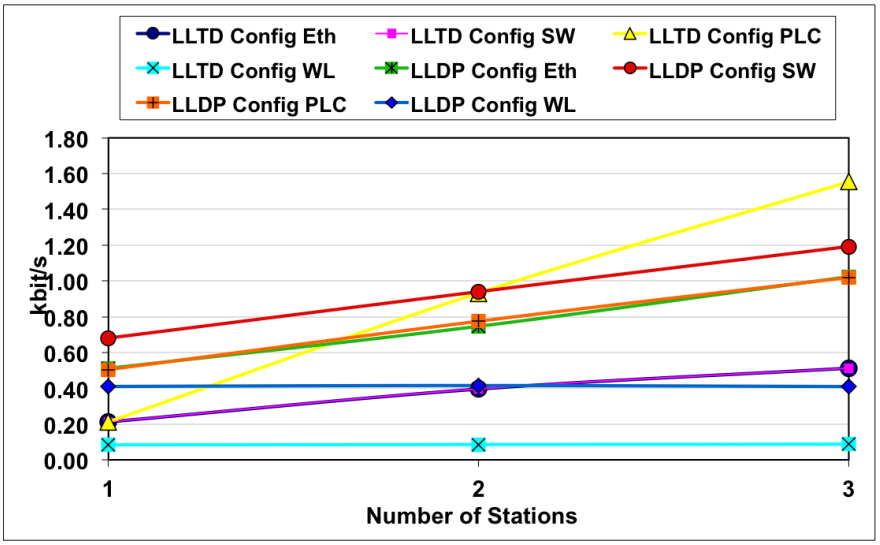


Figure 6.6: *Average injected traffic rate.*

6.5.4 Memory Requirement

In terms of memory usage, we observed that the LLTD mapper requires a minimum of 44 Kbytes and a maximum of 120 Kbytes to operate. The relevant application library is called lltdsvc.dll and its size is 188 Kbytes. In total, LLTD thus requires 232-308 Kbytes of memory. If we use all mandatory and optional TLV types, and assuming a maximum data size for each field, for each neighbor device, an LLDP HNID MIB will require 1446 Bytes to store its information. The size of the LLDP background application used in the testbed is approximately 500 Kbytes. In the worst case scenario (3 stations), an LLDP HNID thus needs approximately 505 Kbytes of total memory. This is nearly double the amount of LLTD, but in both cases this use of memory will not typically disturb normal service operation on the HG (requirement 3).

6.6 Conclusions

To our knowledge, this is the first time that a framework to evaluate the operation and performance of topology discovery protocols for heterogeneous consumer networks is established. We defined five performance indicators, namely device classification accuracy, network graph accuracy, discovery time, average injected traffic rate, and memory use. The classification accuracy defines the correctness of the discovered device types and the graph accuracy indicates the correctness of the discovered link types.

The assessment of LLTD and LLDP showed that, although LLTD performs slightly better than LLDP (mainly in terms of discovery time and graph accuracy), none of the protocols fulfills the service providers' requirements satisfactorily. LLDP relies on the presence of manageable HNIDs. Having to address individual HNIDs with a remote management protocol to read out the internal data bases makes an architecture based on LLDP relatively slow and therefore less suitable for use by service providers. Besides, manageable HNIDs are not yet common within home networks. The main weakness of LLTD is the accuracy of the discovery results for home networks containing collision domains linking more than two HNIDs.

Topology discovery is a monitoring application and does not provide solutions for the detected problems. Methods or systems must be developed to intelligently exploit this information according to the service providers' needs. Examples are the design of an expert system to recognize topologies that could affect QoS, and a logging system to store historic topology information

Although the set of configuration we used for assessing the protocols' performance is based on realistic home-network topologies and is fairly complete, it should be analyzed how much the result differ for less common configurations.

Chapter 7

Traffic Prediction in Home Networks

The quality of Internet services depends crucially upon the performance of the home network. Yet the characteristics of home-network traffic are largely unknown due to the difficulty in performing measurements beyond the home gateway. In this chapter, we show that based on a set of traffic measurements, we have been able to model and, in turn, make predictions of home-network traffic dynamics. We discovered that home-network traffic dynamics is significantly different from the characteristics observed in the public Internet. We also found that it is actually possible to build accurate prediction models based on relatively few observations. Thus, it is theoretically possible to realize network optimizations and QoS control inside the home network without requiring intensive monitoring activities. We illustrate the benefits of our model through an IPTV scenario. The findings of this chapter are currently under review for publication of Delphinanto et al. (2012).

7.1 Introduction

The need for new broadband services such as IPTV, IP telephony and gaming and the convergence between fixed and mobile networks have led to a major upgrade of telecommunication networks all over the world. These new services often place greater transport demands on the network in terms of both quantity and quality. The situation is further complicated by the fact that service providers want to guarantee end-to-end service quality to end-devices in the home (Delphinanto et al., 2009b), considering not only the provider-supported terminals but also the myriad of devices that the customer can buy off-the-shelf. The huge heterogeneity of user's devices in

terms of hardware and software configurations leads to home networks that are extremely unpredictable. Figure 1.1 illustrates a typical home network: it consists of devices supporting many different physical- and link layer technologies (for example wireless LAN, Ethernet, and power-line) and topologies. These devices are interconnected with each other and to the Internet via a home gateway (HG), enabling many different services, which originate both from inside the home and from the public Internet.

In such a heterogeneous context, the service's reliability and performance depend crucially on the characteristics of the home network. Thus, characterizing the home-network traffic is the very first step for a service provider to deliver the required quality of service and deploy optimized services.

However, conducting traffic measurements inside the home is extremely complicated. There are many possible configurations (hardware and software), and many usage patterns and applications. What is more, privacy and regulatory constraints make it extremely difficult to obtain the necessary permissions, both from the network operator and from the users. Under such constraints, collecting representative samples that would lead to usable traffic models is a hard task.

We embarked on our project with the hypothesis that, through a suitable methodology, it was going to be possible to derive traffic-prediction models starting from a very small set of sample measurements. This chapter describes such a methodology. We set off by conducting a home-network monitoring campaign in the Netherlands and managed to develop a Markov model that makes accurate traffic predictions. We adopted various cross-validation techniques to determine the usability of our prediction models. We found that it is possible to accurately characterize home-network traffic dynamics and, thus, realize network optimizations and QoS controls, without requiring intensive monitoring activities. We illustrate the benefits of our model through an IPTV scenario.

The remaining of this chapter is organized as follows. The next section provides an overview of previous work with regard to home-network characterization. We continue to describe our traffic measurement methodology and results in Section 7.3. Further, we discuss the method to build a model that represents the traffic behavior in Section 7.4 and present an in-depth analysis of validation of the model in Section 7.5. Finally, we discuss how the model can improve the performance in IPTV service admission control in Section 7.6.

7.2 State-of-the-Art

The home network contributes to the critical 'last mile' access on the Internet. End-to-end service delivery is particularly susceptible to the stochastic nature of the home

network, whereby different sources of competing traffic often result in disruptions.

One approach has been to address the unreliability of the home network through a rigorous management of the resources by the home gateway, which assures service quality via a dedicated module (Chen et al., 2007; Duenas et al., 2005; Royon & Frenot, 2007). However, to be effective these solutions require significant upgrades in the home network. In any case, quality of service management requires an in-depth knowledge of the network, including its usage patterns and traffic characteristics. The studies performed by DiCioccio et al. (2010) revealed that the home network has indeed a significant impact on service performance end-to-end. However, their study is conducted in a controlled environment, while we base our work on a set of measurements performed on live networks, namely home networks as they are used in real households.

Also relevant to our experiments is the work by Dischinger et al. (2007) that looks at the end-to-end network and, specifically, identifies the different properties of access networks in the USA and Europe, respectively. In comparison, our work captures a much smaller geographical area, i.e. The Netherlands. However, our contribution goes beyond the analysis of high-level properties: we developed a framework that goes from monitoring the network, to characterizing it and, finally, building predictions of traffic behavior.

The importance of prediction models is exemplified by a number of studies that range from traffic engineering to anomaly detection and network diagnostics. Overall, the ability to predict how the traffic will evolve based on the observation of the current state of the network, leads to better resource management. Yuan et al. (2010) have suggested a method that combines statistical data and traffic prediction techniques to maximize spectrum efficiency and quality of service in cognitive radio networks. Cortez et al. (2007) have proposed a method for the prediction of TCP/IP traffic in the Internet. Furthermore, Li et al. (2005) have used traffic predictions to improve the performance of video streaming over wireless LANs. Another exemplary application is in the area of security. Jiang & Papavassiliou (2004) have shown how the characterization of the network's traffic dynamics may be used to prevent network attacks such as mail-bombing attacks or UDP flooding attacks. While many researchers have studied the general framework of network traffic characterization and its applications, the specific domain of home networks has not been explored.

7.3 Characterization of Home-Network Traffic-Rates

7.3.1 The Profile of Home-Network Samples

Our study is based on data collected in 15 households in The Netherlands. This may seem to be a limited sample, but subjective studies aimed at establishing the user's

perception of quality in telecommunications services are considered to be acceptable when at least 15 "well-chosen" samples are considered (ITU Telecom, 2002). In addition, our original intention was to assess whether it was possible to come up with accurate traffic models based on small-scale monitored data. In this way we could satisfy the severe constraints posed by privacy and regulatory issues and develop a method that would work with unobtrusive monitoring. We describe such a method in the rest of the chapter.

The characteristics of the scrutinized households are summarized in Table 7.1. We selected a population that can be classified as "early adopters", i.e. users who tend to adopt state-of-the-art technologies. We assume that prediction models built upon from these networks will be applicable to a much larger section of the total population in the near future.

Table 7.1: *Type of households under scrutiny (15 in total).*

Profile Parameter	**Type**	**Amount**
House	Regular houses (i.e. adjacent, 3-floor houses).	9
	Apartments.	5
	Suburb house.	1
Network type	Cable-based broadband connection.	2 households
	ADSL subscribers broadband connection.	13 households
	Ethernet, Wireless LAN.	14 households
	Ethernet, Wireless LAN, and Powerline.	1 household
Average number of network devices per household	Laptop or PC.	4.6
	Multimedia servers e.g. Network Attached Storage, IPTV Set-Top-Box, Web server.	1.5
	Personal handheld device e.g. PDA, Smarphone, or Internet tablet.	1.2
	Game console.	0.7
	Others.	1.0
User	Families with children (professional).	12 houses
	Families without children (student, professional).	3 houses
	The average number of users per house.	3 persons
	The average daily time spent on network applications.	4 hours

Due to privacy constraints we were not allowed to perform any application-level monitoring. Thus, in order to select 15 households having the most diverse usage profiles, we carried out a questionnaire-based survey. The distribution of the adopted applications is given in Figure 7.1 (specifically, different applications are given versus the percentage of users that adopted them). As expected, everybody is using email

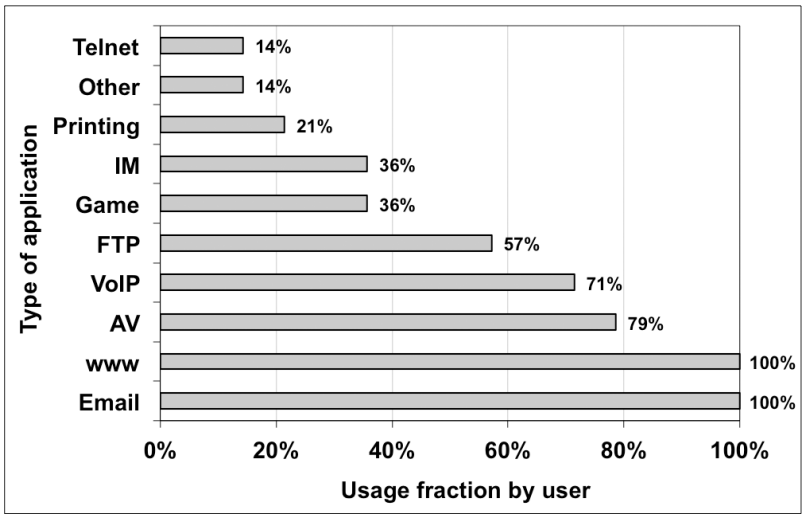


Figure 7.1: *The profile of applications that were used by our home network samples.*

clients and browsers. The least frequently used applications (telnet and other network tools) indicated that about 17% of our users were advanced network users. Relevant to our study is also the fact that 79% of users made use of audio- and video- streaming services, which are representative of data-intensive applications. Gaming is bound to increase significantly in the future, as a lot of online gaming traffic is expected to invade the public Internet. Our sample "online gaming" population included 38% of the total, which is a significant portion in the current Internet panorama. Again this is an indication of our choice to scrutinize "early technology adopters". However, our application-usage survey unveiled that each of our 15 households actually represents a fairly unique case.

7.3.2 The Measurement Set-up

As a home gateway, we adopted the Linksys WRT54 GL v.1.0 broadband router (see Figure 7.2). Our set up allowed the following types of IP traffic flows: incoming from the public Internet into the Local Area Network (LAN); outgoing traffic from LAN to the public Internet; and LAN-to-LAN (namely the home-network traffic). The latter included both 100 Mbit/s Ethernet (ports P1, P2, P3, P5 and P6) and IEEE 802.11g (port P4). P4 was configured in access-point mode. P6 was connected to our IP traffic recorder.

The traffic recorder was intentionally kept separated from the rest of the LAN in order to avoid its interference with the home-network traffic measurements. To achieve

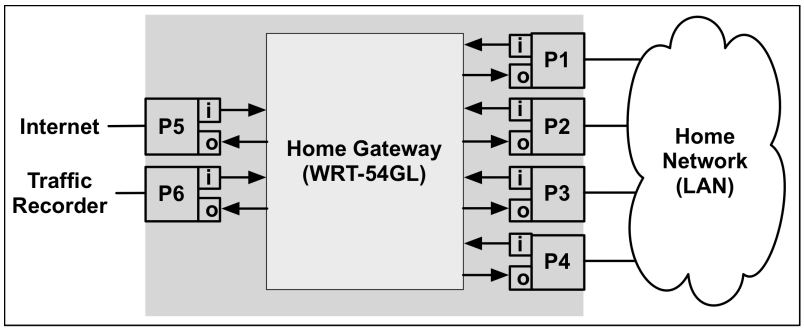


Figure 7.2: *Configuration of our home broadband router. Legend: i = ingoing port; o = outgoing port; P1, P2, P3, P5, P6 are 100 Mbit/s Ethernet interfaces and P4 is an IEEE 802.11g access-point. Port P6 was dedicated to traffic recording.*

accurate measurements, the recorder was bound to incur substantial (non-negligible) traffic to and from the router, which was obviously not included in the home-traffic measurements. The rate of all incoming and outgoing traffic was measured with PRTG[1] at 10 s intervals. It was not possible to adopt smaller observation intervals, which we noticed would be loading the CPU of the router, affecting the normal home-network traffic. Nevertheless a 10 s interval suited to our study, which revealed that the typical traffic-rate fluctuations had a larger periodicity.

We measured the incoming traffic-rate (in bit/s) on all ports except P6 (that was used by the traffic recorder). In each household, there was a different combination of networked devices being connected to ports P1 to P4, using WiFi, Power-line, or Ethernet. The individual traffic rates were based on a 10-s period and were then aggregated using Equation 7.1, whereby n is the measurement sequence (every 10 s); $InTraffic_j(n)$ is the incoming traffic on port-j; $x(n)$ is the total aggregated traffic corresponding to the n-th sequence.

$$x(n) = \sum_{j=1}^{5} InTraffic_j(n) \tag{7.1}$$

We measured $x(n)$ of our fifteen households for seven consecutive days. Let u_q represents $x(n)$ for household q. As $q = [1 \ldots 15]$ then our measurement space is $U = \{u_1 \ldots u_{15}\}$.

7.3.3 Traffic-Rate Characteristics

Figure 7.3 depicts our traffic-rate measurements, expressed as a probability mass function $pmf(x)$, whereby x is the measured traffic rate as given in Equation 7.1. The

[1] PRTG is available at http://www.paessler.com/

bin-size in the graph is set to 1 kbit/s. The $pmf(x)$ is plotted up to 40 kbit/s, while the higher traffic rates are distributed up to 92 Mbit/s (not shown in the figure), with $pmf(x)$ lower than 0.002. The home-network traffic-rate distribution approximates a Generalized Pareto distribution. The approximation has a *Root Mean Square Error* of 0.0005. The home-network traffic-rate distribution differs significantly from that of the public Internet, which is more Gaussian (Pras et al., 2009). This could be because the Internet is always active, while home networks tend to have long periods of relatively low activity. Furthermore, home networks tend to be dominated by a smaller set of applications (Figure 7.1). The $pmf(x)$ of Figure 7.3 also indicates that traffic rates are fairly stochastic and are distributed quite far from their average.

An important consequence of our finding is that many of the existing management and control techniques are not applicable to home networks. An example is the capacity dimensioning method described in Pras et al. (2009), which will only work with networks having Gaussian traffic. To address this problem, we show in the remainder of this chapter that is possible to build traffic prediction models that can, in turn, be used to manage home networks.

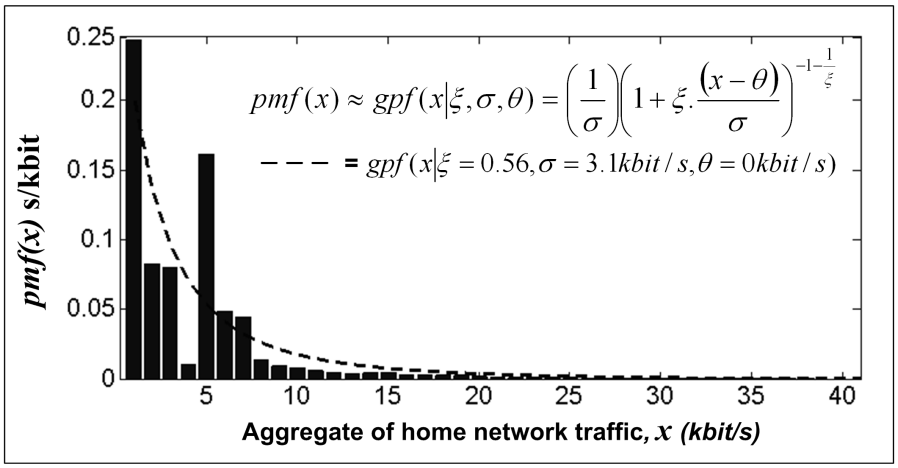


Figure 7.3: *The probability mass function of the aggregated rates of home-network traffic, $pmf(x)$, is approximated with a Generalized Pareto distribution function $gpf(x)$ that has the following parameters: shape (ξ) = 0.56, scale (σ) = 3.1 kbit/s, location (θ) = 0 kbit/s, and average bitrate = 7 kbit/s.*

7.4 Stochastic Model of Traffic Rates in Home Networks

In this section, we build a traffic prediction model for home networks, from measurements in a relatively small set of 15 households. We start off by calculating the *entropy*

period of the observed traffic rates. Entropy is used as a measure of the degree of randomness of the stochastic traffic-rates. We then distinguish various different states of entropy in the home network, and from that we derive a traffic prediction model (namely a Markov transition matrix), which gives the state-transition probabilities of our home-network stochastic traffic-rates.

7.4.1 Entropy Period

The period of entropy is approached by the average of all application session times in the measurement sample. This average time gives an indication of the duration of the use of network applications. During this time period, the user would expect no changes in the network that could lead to disruption of the application. Unfortunately though, due to privacy constraints we were not allowed to monitor the user's application sessions (and this is a typical issue in home network monitoring). To overcome this hurdle we proceeded with an empirical estimation of the session period. This is a delicate process, as any over- and under-estimation would make it impossible to profile the traffic rates.

The procedure for calculating the entropy period is described through Equation 7.2 to Equation 7.5. First, let Δx_n be the variable representing the change between consecutive samples of the traffic-measurements per household:

$$\Delta x_n = |x_{n+1} - x_n|,\ n = 1 \ldots N \tag{7.2}$$

Then, let f be the variable representing the n-th index of Δx_n, when Δx_n is larger than 250 kbit/s, which is the average precision of existing bandwidth tools for home networks (Delphinanto et al., 2011c). Then, f has the following space:

$$F = \{f | f = n\ for\ \Delta x_n \geq 250 Kbit/s\} \tag{7.3}$$

Next, let d be the variable approximating the application session time for an application. Then, d is calculated by taking the distance between f values and has the following space:

$$D = \{d | d = (f_{i+1} - f_i) \cdot 10,\ i = 1 \ldots |F|\} \tag{7.4}$$

Note: in Equation 7.4, $f_i \in F$ and Δf_i is multiplied by 10 because the traffic rate x is measured in 10 s intervals.

Then, we repeated the calculation of Equation 7.2 to Equation 7.4 for all home-network samples, $u \in U$. Consequently, we obtained $D(u)$, namely the space D corresponding to sample u. By aggregating $D(u)$ for all $u \in U$, we obtained $D_{all} = \{D(1) \ldots D(15)\}$. The probability mass function for the application session time $d \in D_{all}$, namely $pmf(d)$, for $d \leq 3000$ s, is depicted in Figure 7.4. The bin-size is set to

100 s. We can observe that about 70% of the application session times were shorter than 500 s, while the remaining ones were spread between 500 s and 8000 s, with a $pmf(d)$ value lower than 0.04. Thus, it would be reasonable to choose an application session time of about 500 s, which captures the majority of the observed samples. Obviously, an entropy period larger than 1000 s could cover a broader set of session times but if we would use such a long period, we may lose the information about traffic-rate fluctuations caused by application of short duration.

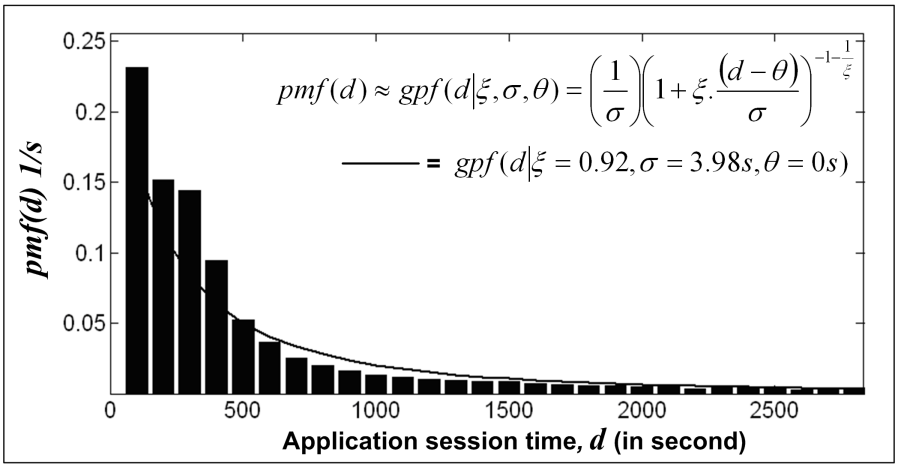


Figure 7.4: *The probability mass function of the approximated application session time, $pmf(d)$ derived from our 15-household measurement campaign. The function can be approximated with a Generalized Pareto distribution $gpf(d)$ having the following parameters: shape (ξ) = 0.92; scale (σ) = 39.8 s; and location (θ) = 0 s. The resulting average application session time is ≈ 500 s.*

The $pmf(d)$ can be approximated with a Generalized Pareto distribution function, namely $gpf(d)$, as shown in Figure 7.4 (solid line). Finally, we defined the entropy period, $T_{entropy}$, as the average of the application session times, as given in Equation 7.5 and obtained a value of about 500 s.

$$T_{entropy} = average(D_{all}) \tag{7.5}$$

This value is consistent with the average user session time reported by Balachandran et al. (2002), in relation to public wireless LANs. They found that about 60% of their samples incurred an average user session time of about 600 s and that the overall distribution followed a Generalized Pareto distribution.

7.4.2 Entropy Calculation

Entropy is a statistical measure of the disorder of a closed system. The use of entropy has been suggested in a number of empirical studies as a means for summarizing the traffic distribution of different applications. For instance, entropy is used for anomalous incident detection (Lakhina et al., 2005) or for traffic clustering and classification (Li et al., 2007). We use entropy as a measure of the degree of randomness of the stochastic traffic-rates. We calculate entropy as follows:

1. For every home-network sample $u \in U$, we measure traffic rates with a periodicity equal to $T_{entropy} = 500$ s. We derive the entropy calculation space, $S(m)$ following Equation 7.6, whereby the index, m indicates the subsequent 500-s intervals and n is the subsequent of the traffic-rate measurement (every 10 s).

$$\begin{array}{lcl} S(m) & = & \{x(n) : n = (50 \cdot (m-1)), (50 \cdot m)\}, \\ m & = & [1, 1209] \end{array} \tag{7.6}$$

2. We calculate the probability mass function of $x(n)$, that is $Pmf(x)$, for every $S(m)$. The bin-size is 1 kbit/s.

3. Let $h(m)$ be the distribution of the entropy values of the traffic rates for $S(m)$, calculated using the following equation.

$$h(m) = -\sum_{x \in S(m)} \left[Pmf(x) \cdot \log(Pmf(x)) \right] \tag{7.7}$$

4. The procedures 1 to 3 were repeated for all home-network measurement samples $u \in U$. Let $h_u(m)$ be the entropy distribution for a sample, u.

5. We normalized $h_u(m)$ with h_{max}, which is the maximum entropy value of all entropy samples. The normalized entropy distribution of sample u, $H_u(m)$ is given by Equation 7.8.

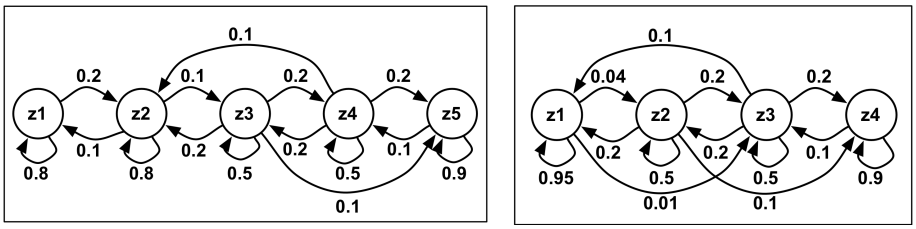
$$H_u(m) = \frac{h_u(m)}{h_{max}},\ h_{max} = \max[h_u(m)]_{u=u_1}^{u_{15}},\ u \in U \tag{7.8}$$

7.4.3 A Markov-Chain Model for Stochastic Traffic-Rates

We use the Markov-chain methodology for developing our model of stochastic traffic-rates in home networks. For this purpose, we first define the possible states and then calculate state transition probability, obtaining the Markov matrix transition.

For the state definition, we initially assumed that the stochastic traffic-rates could be sufficiently represented into five states. The selection of five states was based on intuitive observation of the normalized entropy distribution histogram of Figure 7.4.

If z is the state of the stochastic traffic rates at any m (the state transition step index), the five states are defined by a range of normalized entropy values, namely $z \in \{z_1 = [0, 0.2], z_2 = (0.2, 0.4], z_3 = (0.4, 0.6], z_4 = (0.6, 0.8], z_5 = (0.8, 1]\}$. Consequently, z_1 represents steady traffic-rates (which may be low e.g. during the night, or high e.g. while watching a movie), while z_5 indicates the most stochastic traffic-rates (which suggests many different relatively short-lived applications running in the home network concurrently). The remaining states represent intermediate conditions. We then apply this state definition to the entropy sample $H_u(m)$ to get the state distribution of sample u. We defined this state distribution as $Z_u(m)$. The transition between the states is considered to follow a discrete-time random process with a Markov property, in which the next state depends only on the current state but is independent from previous states. After calculating the probabilities of all state transitions, for all $u \in U$, the relevant Markov-chain can be built, as depicted in Figure 7.5(a).



(a) (b)

Figure 7.5: *The Markov chain for stochastic traffic-rates in home networks, given with five states (a) and then reduced to four states (b).*

Based on the Markov-chain of Figure 7.5(a), we learned that the definition of z_1 was not really useful, since z_1 could only switch to z_2 or to itself. Therefore,, we combined z_1 with z_2. To improve the model, we re-defined the state space, namely $z \in \{z_1 = [0, 0.4], z_2 = (0.4, 0.6], z_3 = (0.6, 0.8], z_4 = (0.8, 1]\}$. Consequently, this yields another Markov chain, as depicted in Figure 7.5 (b).

The Markov chain can be represented as a stochastic matrix, M. The matrix corresponding to the Markov chain of Figure 7.5(b) is expressed in Equation 7.9, whereby each element gives the probability of a specific state transition. Suppose $Pr(z_j|z_i)$ is the probability of the transition from state z_i to state z_j in one step, then $Pr(z_j|z_i)$ is obtained from the i-th row and j-th column element of the matrix. The stochastic matrix M models the behavior of the average scrutinized user (i.e. included in our sample data). The diagonal values in the matrix are relatively large, which indicates that the home-network traffic-rates tend to remain in the same state. This especially counts for z_1 or at z_4.

$$M = \begin{bmatrix} 0.95 & 0.04 & 0.01 & 0 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.2 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix} \tag{7.9}$$

From M we can derive a number of useful conclusions:

1. What is the probability to transit between two possible states after m steps? If $Pr_m(z_j|z_i)$ is the transition probability of state z_i to z_j after m steps, where i and j are either 1,2,3 or 4, then $Pr_m(z_j|z_i)$ is the i-th row and j-th column element of the m-th power of the stochastic matrix, $[M]^m$.

2. How can we predict a future state, given a certain initial condition? Let us assume a time-homogeneous matrix, M (i.e. the matrix remains unchanged after each step). Suppose $Zv(m)$ denotes the state probability vector after m steps from the initial state, Z_{init}. Then, $Zv(m)$ is calculated with Equation 7.10.
   Note: the elements composing $Zv(m)$ are $\{Pr_m(z_1), Pr_m(z_2), Pr_m(z_3), Pr_m(z_4)\}$, where $Pr_m(z) = [Pr(z)]^m$ is the probability of state z after m steps from the initial state $Z_{init}(Z)$.

$$Zv(m) = Z_{init} \cdot (M)^m \tag{7.10}$$

3. How can we calculate long-term probabilities , π (i.e. the steady-state vector), independent from the initial state? Since the matrix M is irreducible (i.e. each state is reachable from any other one and $[Pr(z_j|z_i)]^m > 0$) and has a finite state space, we can obtain π (van Mieghem, 2006):

$$\pi = \{0.61, 0.1, 0.08, 0.21\} \; for \; \pi = \lim_{m \to \infty} Zv(m) \tag{7.11}$$

   This means that, based on the sampled households, the home network are on state z_1 for 61% of the time.

4. How does the use of our model limit the memory load of the home gateway? Thanks to the Markov transition matrix, the HG does not need to keep historical information about the traffic rates, nor the states of stochastic traffic-rates. This beneficial, as the home gateway will not be required to allocate resources to store any historical data.

7.5 Accuracy of the Model

An interesting aspect of our method is its ability to provide accurate predictions even when only a small set of samples is available. This is in fact the typical situation in the context of home networks. The accuracy of our model is assessed in this section.

7.5.1 Model Validation

A cross-validation technique is used for measuring the predictive performance of our model. This technique is widely accepted in the data mining and machine learning community, and serves as a standard procedure for performance estimation (Racine, 1997). Basically, cross-validation divides data samples into two data sets: (1) training sets, used to train or to develop a statistic model; and (2) validation sets, used to test the model. The test is repeated through several iterations and for different combinations of training sets and validation sets, selected from the complete set of samples. Finally, the model's performance is derived from the average result of the individual validation tests. Many cross-validation methods exist, which mainly differ on the procedures followed for composing the training and validation sets. The most relevant methods for our study include:

1. k-fold cross-validation. In this method, the samples are first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed so that (within each iteration) a different fold of the samples is held-out for the validation sets, while the remaining $k-1$ folds are used for the training sets. The samples are commonly stratified prior to being split into k folds. Stratification is the process of rearranging the samples, as to ensure that each fold is a good representative of the whole. Kohavi (1995) recommended that k should be 10 to provide little biased estimations of the performance.

2. Repeated k-fold cross-validation. In this method, the k-fold cross-validation process is repeated multiple times. The data is reshuffled and re-stratified before each round. Consequently, the repetitions increase the reliability of the validation.

3. Leave-one-out cross-validation (LOOCV). This method is a special case of k-fold cross validation, where k equals to the number of samples. In each iteration, all samples except for a single observation are used for training and testing the model on that single observation. The result obtained using LOOCV is known to be almost unbiased. This method is widely used when the available data is limited (Refaeilzadeh et al., 2009).

We decided to use LOOCV for measuring the performance of the stochastic traffic-rate model, because the other cross-validation methods need fairly large samples and suffer from bias estimation. To the best of our knowledge, this method has never been applied on data sets with the Markov property. We applied the LOOCV method using the following procedure:

1. We considered that the fifteen samples of the home-network traffic-rate measurement formed fifteen independent observations. Let u represent the observation sample, where $u \in U = [1, 15]$.

2. The observation samples were partitioned into two data sets: (a) a validation set U_v that contains with one observation sample, or $U_v = \{u\} \subset U$, and (b) a training set U_t that contains the remaining observations, or $U_t = U - U_v$.

3. For building a test set of states, the traffic-rate data in the validation set, U_v was converted to *entropy* distribution, namely $H_{validation}(m)$, as per Equation 7.6 - Equation 7.8. The state definition (Section 7.4.3) was then used to transform $H_{validation}(m)$ to state distribution $Z_{validation}(m)$. The index m represents the sequence of states in the distribution collected for seven days in periods of 500 s. Since $Z_{validation}(m = 1)$ is used for an initial state, the range of m is $2 \ldots 1209$.

4. A Markov transition matrix was built using the training set U_t (see Section 7.4.3).

5. To obtain the state probability vector, $Zv(m)$, the transition matrix is parsed through Equation 7.10, where the initial state was set as $Z_{init} = Z_{validation}(1)$.

6. The accuracy of the model corresponds to the True Positive Rate (TPR) parameter, which gives the number of correct predictions for what will be the next state. Several prediction methods were employed, namely:

   (a) *Random*, this is a case of a random system with four states, where each state has the same probability to appear (all transition probabilities equal to 1/4).

   (b) *Intuitive*, the predicted state is always the state with the highest probability in $Zv(m)$.

   (c) *Dominant*, the predicted state is always the dominant state in the steady state vector. In our case the dominant state is z_1 as the steady state vector of our matrix is $\pi = \{0.61, 0.1, 0.08, 0.21\}$.

   (d) *Selective*, the same as the *Dominant*, but the steady state vector is recalculated for every new model, namely after every iteration.

(e) *Persistent*, the next state is predicted to be always the same as the current state, so $z(m+1) = z(m)$, and when the prediction was wrong the next prediction would be taken from the current state of the test set, or $z(m) = Z_{validation}(m)$. Since our matrix is fairly diagonal this method will give the best predictive performance.

(Note: the *Random* and *Persistent* methods are used only for the worst and the best performance indicator, respectively.)

7. Suppose $TPR(U_v)$ is TPR for validation set for U_v then the final TPR is the average of $TPR(U_v)$.

8. Step 2 to 6 is repeated for all $U_v \subset U$ and the final TPR is calculated as the average of all $TPR(U_v)$.

9. Step 8 is repeated for a smaller number of observation samples, namely 1 to 14 observations and also takes into account its combinations from the available (fifteen) observations. For observation sample number = 1, the validation set is taken the same as the training set.

7.5.2 Prediction Accuracy

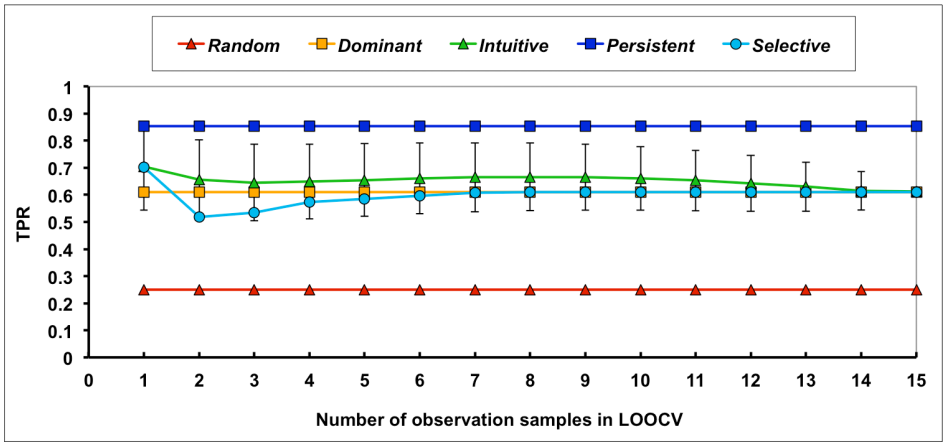


Figure 7.6: *The accuracy (TPR) of several prediction methods used in LOOCV as a function of number of observation samples.*

The results of TPR calculations with different prediction methods are plotted in Figure 7.6. The validation set consists of a sequence of states where as the model tries to reproduce that sequence. Because of the probabilistic character of the model,

the prediction accuracy would usually not hit 100% ($TPR = 1$). The performance of the prediction methods can be ranked based on TPR as follows (the higher TPR the better): *Persistent*>*Intuitive*>*Dominant*>*Selective*>*Random*. The best performance is achieved when *Persistent* is employed, which gives $TPR = 0.85$. Although this technique performs the best, it is not applicable to implementation on the HG because this method requires a real time actual state monitoring for every transition sequence. The worst predictive performance is given by *Random*, that yields $TPR = 0.25$, which means one just guesses blindly what the next state will be. Therefore these two results are the upper and lower limits of our prediction performance.

The *Dominant* method gives a constant prediction performance for all samples, namely $TPR = 0.6$. In general, the *Dominant* method gives a better result than the *Selective* method. This is because that not all samples have z_1 as the dominant state. When a model with a dominant state $= z_1$ is tested on a sample where the major state is not z_1, more incorrect predictions will happen than when the sample's major state is z_1. We noticed that there are three samples where major state is not z_1. The influence of these samples on the average TPR becomes less with bigger LOOCV sample number. With regard to ease of implementation, the *Dominant* method is more preferable than the *Selective* method, it does not need to recalculate the dominant state when the samples change.

The *Intuitive* method is particularly interesting. In general this method gives better predictive performances than *Selective* and *Dominant*. The performance of *Intuitive* decreases from $TPR = 0.70$ to $TPR = 0.61$, when it is applied to more than one household. This is expected since households do not behave in the same way. Furthermore, with a larger sample TPR of *Intuitive* decreases but its error also decreases. With regard to implementation, the *Intuitive* method needs no real-time measurement since the prediction is only based on the traffic model and an initial state. The initial state for *Intuitive* can be defined in several ways, for example: (a) taken from the dominant state, (b) by setting the state to be z_1 or z_2 during a quiet period, or vice versa, or (c) a random state since in the long run the prediction model will converge to a steady state.

Despite the implementation issues, all prediction techniques give a prediction accuracy that is much better than *Random*. When considering prediction performance and implementation benefit, *Intuitive* gives the best prediction performance, namely 72% from the optimal predictive performance, given by *Persistent*. Several other methods can be employed to improve the prediction accuracy (higher TPR), such as segmentation of households with the same behavior, or applying a self-learning machine to update the transition matrix while adapting to new user behaviors.

7.6 Use Case: the IPTV Service Admission

IPTV services are sensitive to packet loss and delay, especially in an unreliable link. The IPTV stream has strict minimum bandwidth requirements to facilitate the right number of frames per second for delivering moving pictures. For example, an MPEG-2 compression method-based IPTV service needs 2 to 4 Mbit/s for a standard definition television (SDTV), 8 to 10 Mbit/s for a DVD-quality, and 18 to 24 Mbit/s for a high-definition television (HD-TV). This is set of requirements is needed to guarantee the quality of user experience. For meeting the requirements, service providers often provide a proprietary device, namely a set-top-box (STB) to be connected directly to the TV, behind the home gateway (HG), see Figure 1.1. Consequently the STB requires an exclusive connection to the gateway and leaves no option for interconnection to the rest of home network. This is mainly because the service providers consider home networks to be heterogeneous and unreliable, and potentially jeopardizing the quality of service. The HG now could solve the quality of service problem. This is where our model of stochastic traffic-rates plays a role. This role is depicted with the flowchart in Figure 7.7.

1. Upon an IPTV service request, the HG will ask for the IPTV bandwidth requirement (i.e. $A_{required}$) and a planned duration of consumption (i.e. $T_{required}$ in second). This information can be provided by a user input or given by standard set values.

2. To meet $A_{required}$, the HG needs to measure the real-time available bandwidth, namely $A_{measured}(m=0)$, between itself and any useful end-device (for example network TV or STB), where m is an index of time interval. Available bandwidth (AB) measurement techniques for home networks are available but they cannot guarantee that $A_{measured}(m)$ will be the same in the future for $m+1$ (Delphinanto et al., 2011c; Li et al., 2008). A direct consequence is to measure $A_{measured}(m)$ very often such as in Lee et al. (2006). If more similar services are running concurrently; the home network may be flooded with AB measurement traffic: a single AB measurement can generate $\sim$ 1.5 Mbit/s traffic (Delphinanto et al., 2010).

3. The benefit of the stochastic traffic-rate model is that we can predict the future behavior of the traffic in the home network. Suppose M and Z_{init} are provided (see Equation 7.10) and the *Intuitive* prediction method is used (see Section 7.5) then we can predict the future traffic, namely $z(m)$.

4. Because the IPTV service is planned for $T_{required}$, which is typically long, the value of $T_{required}$ will contain a number of state transitions, namely $N_{required}$,

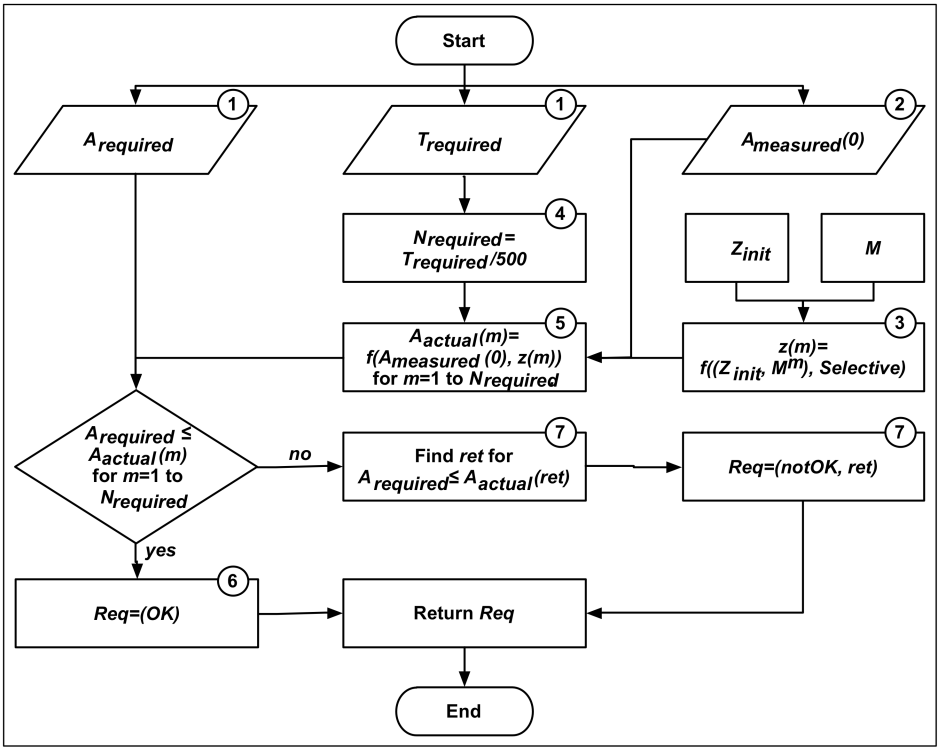


Figure 7.7: *A flowchart showing the use case of how the stochastic traffic-rate model can improve an IPTV service admission.*

departing from the current state m, where $N_{required} = T_{required}/500$ s (assuming each transition is 500 s long).

5. By knowing $A_{measured}(m = 0)$ and the future state, namely $z(m)$, the effective available bandwidth from $m = 1$ to $N_{required}$, namely $A_{actual}(m)$, can be calculated. Thus $A_{actual}(m)$ is a function of $A_{measured}(m = 0)$ and $z(m)$. Principally, the function corrects $A_{measured}(m = 0)$ with a safety margin that is as a result of stochastic traffic-rates predicted at state $z(m)$.

6. HG can check whether $A_{actual}(m)$ can meet $A_{required}$ for $m = 1$ to $N_{required}$. If in (nearly) all of the conditions $A_{actual}(m) \geq A_{required}$ then the IPTV service can be started.

7. If in too many of the intervals (i.e. for $m = 1$ to $N_{required}$) $A_{actual}(m) \leq A_{required}$, then HG can advice to decline the IPTV request (namely $Req = notOK$) and suggest to retry the request after ret periods. ret is calculated from the as being

the period after which the available bandwidth is predicted to fulfill the requirement. Or the user can be suggested to turn off some other applications to allow more available bandwidth.

To give a quantitative illustration of the case above, suppose that the IPTV service is requested for one hour. Then eight bandwidth measurements are required without the presence of the prediction model with 500 s interval between the measurements. This is in contrast with only one measurement if the model is employed. In this case, the model can reduce the number of AB measurement is by a factor 8. This saving can be even more significant when many similar applications are present in the home network. Another direct consequence of suppressing the number of measurements is that the HG's processing power can be saved. Thus, the prediction model can help in providing quality of service for IPTV using little resources from the HG.

7.7 Conclusions

Traffic characterization and traffic prediction are very useful techniques in computer networks. The main research question in this work is if it is possible to characterize and then to build a prediction model of home-network traffic. The main challenges in this work are that home networks are technologically heterogeneous and access to the home is very limited due to privacy issues. It is difficult and expensive to perform a long traffic monitoring experiment in many home networks to collect samples that are significant enough for the ability to build a prediction model. What we found from this work is that it is possible to characterize the home-network traffic dynamics based on relatively few samples. Home-network traffic is observed as unique and different from any other kind of networks, such as the broad Internet. From the traffic characterization experiments, we also managed to build a traffic prediction model for the home network, which is accurate enough for the purpose of network optimization and quality of service management. Our prediction model is in fact close enough 72% to the optimal prediction accuracy to be useful in practice. As a result of this work, we can now apply network optimization and quality of service control inside the home network without requiring intensive monitoring activities, but by simply using the prediction model in combination with actual monitoring by probing the network a few times per hour.

Chapter 8

Final Remarks

This chapter summarizes the main findings of our research and draws their implications on network and service monitoring research, specifically in heterogeneous home networks. It thereby pertains to answer the research questions formulated in Chapter 1. The experimental studies, the implementations and the evaluations underline each of the answers as conclusion of the research work. After highlighting these findings, the chapter poses some directions for future developments and research in home networks.

Despite the grown interest in home networking, the technologies of home network remain extraordinarily difficult for people to install, manage, and use. The thesis developed an array of answers using engineering and analysis work to the challenge for near future home networking. The work includes combinations of empirical and field studies to understand the use of the developed protocols, models and architectures of the domestic setting of communication network. We expect that the research results can give a general idea to point in the direction of near future home-network techniques that inform the users of the implications of network changes, and -infrastructures that are able to configure and repair themselves.

1. How can we enable a seamless interaction between devices that use different network technologies and support different service discovery protocols, without adding more complexities to existing state-of-the-art?
Proxy servers can extend the range of service- and device discovery protocols to non-IP domains, such as the Personal Area Network or the car network. We developed a proxy server that enables UPnP devices and services to be discovered on the Bluetooth network and vice versa allowing Bluetooth devices and UPnP control points to control services located on devices in the IP network and the *piconet*, respectively. All functionality needed for effective proxying can be kept within a single logical and/or

physical device. No additional stacks or software will be needed on the other devices in the *piconet* (or IP network). This enables the discovery of not only complex devices, but also very simple ones, such as headsets and toys. The proxy does not demand more processing resources than the ones currently available on mobile phones. To reach maximum interoperability, we suggest that the proxy implements standardized services that have the same capabilities. However, even then a number of standard services in one SDP may not have clearly corresponding services in another SDP, and vice versa. As a consequence, the match between the two different SDPs is sometimes impossible, or only possible for part of the functionality. The interoperability may also cause a downgraded performance, especially for the more superior SDPs. For example, the proxy UPnP CDS and Bluetooth FTP reduce browse time and data throughput to about 50% of the bare Bluetooth and UPnP performance.

In the near future, the proxy can be used for all applications that extend service discovery to non-IP private networks such as in Personal Networks (PNs) (Niemegeers & de Groot, 2002; Freeband.nl, 2008). A PN connects all private networks of a single user seamlessly. If we can interconnect the IP-based private networks by smart routing, then we only need the proxy to realize automatic device- and service discovery over the whole PN, including the non-IP parts. In longer term, explorations should be carried out into promoting prospective interoperability between similar services, and integration of multiple networking and platform technologies through a series of gateway and shared devices.

2. How can we provide a generic solution for SDPs in providing information on the actual availability of (device) resources in the network so the resource conflicts can be avoided, without requiring any modification of current standards or devices in the home?

The resource conflict in SDP can be mitigated with a manageable resource reservation manager that can generally work in every SDP, and can potentially manage resource reservation across SDPs. Therefore, the residential gateway seems to be suitable for hosting the reservation manager. Also the fact that current gateways already have remote management functionality makes them attractive. We developed such resource reservation manager based on common operation of existing SDPs. This does not require modification of current standard networked devices and is backward compatible. However, the reservation manager concept does require clients to cooperate. In the case of UPnP, the reservation manager introduces only a limited amount of extra control traffic on the network and a relatively small reservation application delay. Nevertheless, it can reduce the number of conflict incidents drastically. Moreover it could mitigate the bandwidth occupation significantly and raise the system scalability.

As future work, the resource reservation can be improved by considering contexts such as quality of service, security and privacy. For example, the resource reservation

can be combined with current QoS technologies that are based on traffic differentiation (van Hartskamp et al., 2006). Service access can then be made dependent on the current state of the network. Also, the traffic priority can be made dependent on service availability.

3. How can we provide a generic solution for remote discovery and management of end-devices in heterogeneous home networks, with minimal modifications to existing devices and remote management servers?

We proposed a mechanism for the remote discovery and representation of end-user devices in the home network. The main novelty of the mechanism is that it enables the HG to discover uniquely various types of end devices, using four different discovery techniques concurrently, and to communicate the discovered information to the remote management server in a single data model object. It provides a simple overview of devices in the private network and it prevents contradictory management actions. We also described an architecture and a proof-of-concept for the remote management of UPnP devices with a TR-069/UPnP proxy on the HG. The main advantage of the chosen architecture is the spread of intelligence over the system. The ACS only needs limited extensions and does not need to function as a control system instead of a management server. The proxy implementation is very simple. The end devices may need an extra remote management service to be added, but not another protocol suite. Furthermore, this architecture is basically generic for any combination of web-services like protocols, such as Open Mobile Alliance - Device Management (OMA-DM), Device Profile for Web Services (DPWS) and proprietary implementations. The proof-of-concept works well and will be used to test the qualitative assessment of the scalability of our architecture in a more quantitative way.

The future works may include the remote device discovery of non-IP based devices such as Bluetooth, Zigbee, etc. and the remote management of devices that support other SDPs or web-services like protocols. It is also important to investigate various scenarios upon failures of the remote management. For example, when the HG or the ACS fail what will be the relevant recovery solution?

4. How can home-network topology discovery protocols contribute to the service provider's remote management needs?

We introduced a framework to evaluate the operation and performance of topology discovery protocols for heterogeneous consumer networks. The framework consists of five performance indicators, namely device classification accuracy, network graph accuracy, discovery time, average injected traffic rate, and memory use. We applied the assessment framework to LLTD and LLDP. The results showed that, although LLTD performs slightly better than LLDP (mainly in terms of discovery time and graph accuracy), none of the protocols fulfills the service providers' requirements satisfactorily.

Topology discovery is a monitoring application and does not provide solutions for the detected problems. Methods or systems must be developed to intelligently exploit this information according to the service providers' needs. Examples are the design of an expert system to recognize topologies that could affect QoS, and a logging system to store historic topology information.

Some elaboration of this work may include: to design and test a new topology discovery architecture which can address all service providers' requirements successfully; to research a topology protocol that includes non-IP network domains, such as Zigbee; and to evaluate ITU-T's HTIP when the standard is completed and implemented. Furthermore, the set of configurations we used for assessing the protocols' performance is based on realistic home-network topologies and is fairly complete, but it should be analyzed how much the results differ for less common configurations

5. How can we estimate path capacity and available bandwidth for heterogeneous home networks in a way that requires minimal modification to the end-devices, which is not intrusive to the network, and is accurate enough for the most relevant home applications?

We proposed a technique, so-called *Allbest*, for estimating the path capacity and available bandwidth in heterogeneous home networks. *Allbest* is based on the lightly intrusive packet-pair dispersion technique introducing five main innovations. First, it uses round-trip-time measurements without requiring any modifications to the existing client devices in the home, whilst supporting asymmetric media. Second, it has a fast convergence and an acceptable accuracy, enough to make decisions about the admission of IPTV-like application-traffic streams. Third, it is applicable for a network path that consists of different link-layer technologies and requires hardly any pre-knowledge of the home-network link-layer topology. Fourth, for the path capacity estimation it has two varieties based on ICMP Ping/Echo and UDP/ICMP Error respectively, which in concatenation are able to measure forward and reverse path capacities in asymmetric networks. Fifth, to the best of our knowledge, *Allbest* is the first PGM tool that can estimate the capacity and available bandwidth with the same probing packets. This besides reduces the measurement time but also the traffic being generated.

In the future, *Allbest*'s performance can be investigated on many different network configurations and parameters. Especially, new network technologies supporting IP are currently entering the home such as HomePlug, MoCa, IEEE 1901, IEEE 802.11n, and G.hn. Since some of them (e.g. HomePlug) are exhibiting different physical- and link layer properties (such as fast rate adaptation) from the networks studied in this research, the *Allbest* method needs to be improved to include these novel techniques. The applicability of *Allbest* to Wide Area Networks is also interesting to investigate; next to other consumer networks such as In-car networks, personal area networks, ho-

tel networks, etc., which exhibit similar properties and management issues to home networks. Increasing the accuracy of *Allbest* up to a level that it can be used for Voice-over-IP services or yielding information on other relevant QoS parameters such as delay and packet error rate will be also challenging.

6. Is it possible to characterize home-network traffic and then build a prediction model for it?

It is possible to characterize the home-network traffic dynamics based on relatively few samples. Home-network traffic is observed as unique and different from traffic in any other kind of networks, such as the broad Internet. From the traffic characterization experiment, we also managed to build a traffic prediction model for the home network, which is accurate enough for the purpose of network optimization and quality-of-service management. Our prediction model is in fact close enough to the optimal prediction accuracy (72%) and is therefore practical. As a result of this work, we can now apply network optimizations and quality of service control inside the home network without requiring intensive monitoring activities, but by simply using the prediction model in combination with actual monitoring by probing the network a few times per hour.

For future work it will be interesting to further improve the prediction model's accuracy and establish which other applications can benefit from the model.

Bibliography

Alanqar, W. & Jukan, A. (2004). Extending end-to-end optical service provisioning and restoration in carrier networks: opportunities, issues, and challenges. *IEEE Communication Magazine*, **42**(1), 52–60.

Allard, J., Chinta, V., Gundala, S., , & Richard III, G. G. (2003). Jini meets UPnP: An architecture for Jini/UPnP interoperability. In *Proc. of Symposium of Applications and the Internets*, pages 268–275.

Atinav Inc. (2006). AveLink technology framework. http://www.atinav.com.

Ayyagari, A., AbiEzzim, S., & Zintel, W. (2001). Proxy-bridge connecting remote users to a limited connectivity network, US 2001/0033554 A1, United States Patent.

Balachandran, A., Voelker, G. M., Bahl, G. P., & Rangan, P. (2002). Characterizing user behavior and network performance in a public wireless LAN. *ACM SIGMETRICS Performance Evaluation Review*, **30**, 195–205.

Balemans, H., Smedt, A. D., den Hartog, F. T. H., & Onnegren, J. (2006). Concurrent remote management of CPE by multiple service providers. In *Proc. of BroadBand Europe 2006*, Geneva, Switzerland.

Bellovin, S. M. (1992). A best-case network performance model. Technical report, ATT Research.

Bettstetter, C. & Renner, C. (2000). Comparison of service discovery protocols and implementation of the service location protocol. In *Proc. of the Sixth EUNICE Open European Summer School*.

Black, R., Donnelly, A., & Fournet, C. (2004). Ethernet topology discovery without network assistance. In *Proc. of the 12th IEEE International Conference on Network Protocols (ICNP 2004)*, pages 328–339, Berlin, Germany. IEEE Press.

Bluetooth.com (2001). Bluetooth ESDP for UPnP. Technical report, Bluetooth SIG, http://www.bluetooth.com/pdf/ESDP_UPnP_0_95a.pdf.

Bluetooth.org (2007). Bluetooth, http://www.bluetooth.org.

Botta, A., Dainotti, A., & Pescape, A. (2007). Multi-protocol and multi-platform traffic generation and measurement. In *IEEE Infocom 2007, Demo Session*, Anchorage, Alaska, USA.

Broadband-forum.org (2006a). Data model template for TR-069 enabled device, TR-106 amendment 1. Technical report, http://www.broadband-forum.org.

Broadband-forum.org (2006b). Internet gateway device data model for TR-069, TR-098 amendment 1. Technical report, http://www.broadband-forum.org.

Broadband-forum.org (2006c). TR-126: Triple-play services Quality of Experience (QoE) requirements. Technical report, http://www.broadband-forum.org/technical/download/RT-126.pdf, December.

Broadband-forum.org (2007). CPE WAN management protocol v1.1, TR-069 amendment 2. Technical report, http://www.broadband-forum.org.

Castellanos, E. D. (2010). Characterizing Dutch home network dynamics. Technical report, TNO Internship Report, Delft University of Technology.

Castellanos, E. G. D., Delphinanto, A., & den Hartog, F. (2012). Performance analysis of home network topology discovery protocols. In *Proc. of the IEEE Consumer Communication and Networking Conference (CCNC 2012)*, Las Vegas, NV, USA.

Chen, J.-L., Chen, M.-C., & Chian, Y.-R. (2007). QoS management in heterogeneous home networks. *Computer Network*, **51**, 3368–3379.

Cortez, P., Rio, M., de Sousa, P. N. M., & Rocha, M. (2007). Topology aware internet traffic forecasting using neural networks. In J. M. de Sá, L. A. Alexandre, W. Duch, and D. P. Mandic, editors, *Proc. of ICANN (2), Artificial Neural Networks*, volume 4669 of *LNCS*, pages 445–454. Springer.

de Rocha, A., Leao, R. M. M., & Silva, E. S. (2007). An end-to-end technique to estimate the transmission rate of an IEEE 802.11 WLAN. In *Proc. of the 2007 IEEE International Communication Conference (ICC 2007)*, pages 415–420, Glasgow, UK.

Delphinanto, A., Lukkien, J. J., Koonen, A. M. J., Madureira, A., Niemegeers, I. G. G. M., den Hartog, F. T. H., & Selgert, F. (2007a). Architecture of a bidirectional Bluetooth-UPnP proxy. In *Proc. of the IEEE Consumer Communications and Networking Conference (CCNC 2007)*, pages 34–38, Las Vegas, NV, USA. IEEE Press.

Delphinanto, A., Koonen, A. M. J., Peeters, M. E., & den Hartog, F. T. H. (2007b). Proxying UPnP service discovery and access to a non-IP Bluetooth network on a mobile phone. In *Proc. of the IEEE Symposium of Communications and Vehicular Technology in the Benelux (SCVT 2007)*, pages 1–5, Delft, the Netherlands. IEEE Press.

Delphinanto, A., den Hartog, F. T. H., & Koonen, A. M. J. (2008). Improving quality of experience by adding device resource reservation to service discovery protocols. In *Proc. of the IEEE International Conference on Communications (ICC 2008)*, pages 1813 –1818, Beijing, China. IEEE Press.

Delphinanto, A., Madureira, A., & den Hartog, F. T. H. (2009a). Personal networks proxy-bridge for connecting different types of devices, U.S. 20090303926, United States Patent. 10 december. Granted.

Delphinanto, A., Hillen, B. A. G., Passchier, I., van Schoonhoven, B. H. A., & den Hartog, F. (2009b). Remote discovery and management of end-user devices in heterogeneous private networks. In *Proc. of the IEEE Consumer Communications and Networking Conference (CCNC 2009)*, pages 1–5, Las Vegas, NV, USA. IEEE Press.

Delphinanto, A., Koonen, A. M. J., Zhang, S., & den Hartog, F. T. H. (2010). Path capacity estimation in heterogeneous, best-effort, small-scale IP networks. In *Proc. of the IEEE Conference on Local Computer Networks (LCN 2010)*, pages 1076–1083, Denver, CO, USA. IEEE Press.

Delphinanto, A., Koonen, A. M. J., & den Hartog, F. T. H. (2011a). End-to-end available bandwidth probing in heterogeneous IP home networks. In *Proc. of the IEEE Consumer Communication and Networking Conference (CCNC 2011)*, pages 431–435, Las Vegas, NV, USA. IEEE Press.

Delphinanto, A., Nicolai, F., & den Hartog, F. T. H. (2011b). Network transmission capacity measurement, the Netherlands, WIPO Patent WO/2011/008090. 10 january. Granted.

Delphinanto, A., Koonen, A. M. J., & den Hartog, F. T. H. (2011c). Real-time probing of available bandwidth in home networks. *IEEE Communication Magazine*, **49**, 134–140.

Delphinanto, A., Koonen, A. M. J., Zhang, S., & den Hartog, F. T. H. (2011d). Real-time probing of end-to-end capacity and available bandwidth in heterogeneous local networks. In *Proc. of the IEEE Consumer Communication and Networking Conference (CCNC 2011)*, pages 828–829, Las Vegas, NV, USA. IEEE Press.

Delphinanto, A., Castelanos, E. D., Liu, B., Liotta, A., Koonen, A. M. J., & den Hartog, F. T. H. (2012). Traffic prediction in home network. *Journal of Network and Systems Management. Submitted.*

den Hartog, F. T. H. & Delphinanto, A. (2010). A renewed plea for active home network probing. Contribution HGI01452 to Home Gateway Initiative (HGI), intended for HGI01452. 25 march.

den Hartog, F. T. H. & Delphinanto, A. (2011). UDP throughput and packet loss measurements of HomePlug, contribution HGI01865 to Home Gateway Initiative (HGI). 16 october.

den Hartog, F. T. H., Balm, M., de Jong, C. M., , & Kwaaitaal, J. J. B. (2004). Convergence of residential gateway technology. *IEEE Communication Magazine*, **42**(1), 138–143.

den Hartog, F. T. H., Blom, M. A., Lageweg, C. R., Peeters, M. E., Schmidt, J. R., van der Veer, R., de Vries, A., van der Werff, M. R., Tao, Q., Veldhuis, R. N., Baken, N. H. G., & Selgert, F. (2007). First experiences with personal networks as an enabling platform for service provider. In *Proc. of Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2007)*, pages 1–8, Philadelphia, USA.

DiCioccio, L., Teixeira, R., & Rosenberg, C. (2010). Impact of home networks on end-to-end performance: controlled experiments. In *Proc. of the 2010 ACM SIGCOMM workshop on Home networks*, HomeNets 2010, pages 7–12, New York, NY, USA. ACM.

Dischinger, M., Haeberlen, A., Gummadi, K. P., & Saroiu, S. (2007). Characterizing residential broadband networks. In *Proc. of the 7th ACM SIGCOMM conference on Internet measurement*, IMC 2007, pages 43–56, New York, NY, USA. ACM.

Dobrev, P., Famolari, D., Kurzke, C., & Miller, B. A. (2002). Device and service discovery in home network with OSGi. *IEEE Communication Magazine*, **40**, 86–92.

Duenas, J. C., Ruiz, J. L., & Santillan, M. (2005). An end-to-end service provisioning scenario for the residential environment. *IEEE Communication Magazine*, pages 94–100.

Elisson, C. (2003). UPnP security ceremonies version 1.0, design document. Technical report, UPnP forum.

Figaro Project (2011). Figaro: Future internet gateway-based architecture of residential networks version 1.0. theme [ICT-2009.1.1] The Network of Future. grant agreement no. 258378. Technical report, European Union Project - Figaro.

Freeband.nl (2008). Freeband, http://pnp2008.freeband.nl.

Gratton, D. A. (2002). *Bluetooth Profiles: The Definitive Guide*. Prentice Hall, 1 edition.

Havi.org (2001). HAVi specification version 1.1. Technical report, http://www.havi.org.

HGI (2006). Requirements release 1. Technical report, Home Gateway Initiative, http://www.homegateway.org.

HGI (2008). Home gateway technical requirements residential profile version 1.0. Technical report, http://www.homegateway.org.

IETF.org (2002). RFC3416 and reference therein. Technical report, http://www.ietf.org.

Internet World Stats (2011). Internet usage world stats - internet and population statistics.

ITU Telecom (2002). Methodology for the subjective assessment of the quality of television pictures. recommendation ITU-R BT. 500 11. Technical report, Standardization Sector of ITU.

Jacobson, V. (1988). Congestion avoidance and control. In *Proc. of ACM Symposium on Communications architectures and protocols*, SIGCOMM 1988, pages 314–329, New York, NY, USA. ACM.

Ji, B., Rao, A., Lee, M., Latchman, H. A., & Katar, S. (2004). Multimedia in home networking. *Information Technology, System Application*, **1**, 397–404.

Jiang, J. & Papavassiliou, S. (2004). Detecting network attacks in the internet via statistical network traffic normality prediction. *Journal of Network and Systems Management*, **12**(1), 51–72.

Jini.org (2007). Jini, http://www.jini.org.

Jun, S. M. & Park, N. H. (2004). Controlling non IP Bluetooth devices in UPnP home network. In *Proc. of the 6th International Conference on Advanced Communication Technologies*, volume 2, pages 714–718.

Kampichler, W. & Goeschka, K. M. (2003). On measuring quality of service limitations in local area networks. In *Proc. of the IEEE International Conference on Communications (ICC 2003)*, volume 1, pages 291–295, Anchorage, USA. IEEE Press.

Kapoor, R., Chen, L.-J., Lao, L., Gerla, M., & Sanadidi, M. Y. (2004). CapProbe: a simple and accurate capacity estimation technique. *SIGCOMM Computer Communication Review*, **34**, 67–78.

Keshav, S. (1991). A control-theoretic approach to flow control. *SIGCOMM Computer Communication Review*, **21**, 3–15.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. of the 14th international joint conference on Artificial Intelligence - Volume 2*, pages 1137–1143, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Konno, S. (2008). Cyberlink for Java: Development package for UPnP devices. Technical report, Cybergarage, http://www.cybergarage.org/twiki/bin/view/Main/CyberLinkForJava.

Koponen, T. & Virtanen, T. (2004). A service discovery: A service broker approach. In *Proc. of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004) - Track 9 - Volume 9*, HICSS 2004, pages 90284.2–, Washington, DC, USA. IEEE Computer Society.

Lakhina, A., Crovella, M., & Diot, C. (2005). Mining anomalies using traffic feature distributions. In *Proc. of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM 2005, pages 217–228, New York, NY, USA. ACM.

Lea, R., Gibbs, S., Dara-Abrams, A., & Eytchison, E. (2000). Networking home entertainment devices with HAVi. *Computer*, **33**, 35–43.

Lee, H., Moon, S., Kim, J., & Joe, D. (2006). UPnP-based QoS agent for QoS-guaranteed streaming service in home networks. In *Proc. of IEEE Consumer Communications and Networking Conference (CCNC 2006)*, pages 543–547, Las Vegas, NV, USA. IEEE Computer Society Press.

Li, M., Li, F., Claypool, M., & Kinicki, R. (2005). Weather forecasting: predicting performance for streaming video over wireless LANs. In *Proc. of the international workshop on Network and operating systems support for digital audio and video*, NOSSDAV 2005, pages 33–38, New York, NY, USA. ACM.

Li, M., Claypool, M., & Kinicki, R. (2008). WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks. In *Proc. of the IEEE Conference on Local Computer Networks (LCN 2008)*, pages 374–381, Montreal, Canada. IEEE Computer Society Press.

Li, Z., Yuan, R., & Guan, X. (2007). Traffic classification - towards accurate real time network applications. In *Proc. of the 12th international conference on Human-computer interaction: applications and services*, HCI 2007, pages 67–76, Berlin, Heidelberg. Springer-Verlag.

Limam, N., Ziembicki, J., Ahmed, R., Iraqi, Y., Li, D. T., Boutaba, R., & Cuervo, F. (2007). OSDA: Open service discovery architecture for efficient cross-domain service provisioning. *Computer Communications*, **30**(3), 546–563.

Lowekamp, B., O'Hallaron, D., & Gross, T. (2001). Topology discovery for large ethernet networks. In *Proc. of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM 2001, pages 237–248, New York, NY, USA. ACM.

Madureira, A. (2006). *Architecture of a bi-directional Bluetooth-UPnP proxy*. Master's thesis, Technical University of Delft.

Melander, B., Bjorkman, M., & Gunninberg, P. (2000). A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM 2000)*, volume 1, pages 415–420, San Francisco, USA.

Meshkova, E., Riihijärvi, J., Petrova, M., & Mähönen, P. (2008). A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. *Computer Network*, **52**, 2097–2128.

Microsoft.com (2010). Link layer topology discovery protocol specification. Technical report, http://msdn.microsoft.com/en-us/windows/hardware/gg463024.

Mocalliance.org (2007). Telco TV home networking technology and market outlook: Independent data and analysis by s2 data corporation. Technical report, Multimedia over Coax Alliance, http://www.mocalliance.org. January 2007.

Murty, R., Padhye, J., Chandra, R., Chowdhury, A. R., & Welsh, M. (2008). Characterizing the end-to-end performance of indoor power-line networks. Technical report, Harvard University.

Neisse, R., Vianna, R., Granville, L. Z., Almeida, M. J. B., & Tarouco, L. M. R. (2004). Implementation and bandwidth consumption evaluation of SNMP to web services gateways. In *Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, pages 715–728, Seoul, Korea.

Nicolai, F. (2009). *Capacity measurement in small-scale, heterogeneous, best-effort IP networks*. Master's thesis, Delft University of Technology.

Niemegeers, I. G. & de Groot, S. M. H. (2002). From personal area networks to personal networks: A user oriented approach. *Wireless Personal Communications*, **22**, 175–186.

Nikolaidis, A., Papastefanos, S., Doumenis, G., Stassinopoulos, G., & Drakos, M. (2007). Local and remote management integration for flexible service provisioning to the home. *IEEE Communication Magazine*, **45**(10), 130–138.

Peng, H., Heng, P., Xiangdong, L., & Qiusheng, Z. (2010). Physical topology discovery based on spanning tree protocol. In *Proc. of the International Conference on Computer Application and System Modeling 2010*, pages V14–308–V14–311, Taiyuan,China.

Pras, A., Nieuwenhuis, L., van de, R. M., & Mandjes, M. (2009). Dimensioning network links: A new look at equivalent bandwidth. *IEEE Network*, **23**(2), 5–10.

Prasad, R., Dovrolis, C., Murray, M., & Claffy, K. (2003). Bandwidth estimation: metrics, measurement techniques and tools. *IEEE Network*, **17**(6), 27–35.

Presser, A., Langille, G., Shults, G., Ritchie, J., Walker, M., Kim, C., Ahn, S., Hori, M., Ma, M., Unverferth, J., Bronnenberg, W., Knapen, G., Berkoff, R., Shen, I., Kikkawa, N., Tourzan, J., & Morioko, Y. (2006). Contentdirectory:2 service template version 1.01. Technical report, UPnP forum.

Racine, J. (1997). Feasible cross-validatory model selection for general stationary processes. *Journal of Applied Econometrics*, **2**, 169–179.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In L. Liu and M. T. Özsu, editors, *Encyclopedia of Database Systems*, pages 532–538. Springer US.

Richard, G. G. (2000). Service advertisement and discovery: Enabling universal device cooperation. *IEEE Internet Computing*, **4**, 18–26.

Ritchie, J. (2002). MediaRenderer:1 device template version 1.01. Technical report, UPnP Forum, http://www.upnp.org/specs/av/UPnP-av-MediaRenderer-v1-Device.pdf.

Rosenberg, J. (2009). Obtaining and using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP), RFC 5627. Technical report, IETF Networking working group.

Royon, Y. & Frenot, S. (2007). Multiservice home gateways: Business model, execution environment, management infrastructure. *IEEE Communication Magazine*, **45**(10), 122–128.

Sundramoorthy, V., Scholten, H., Jansen, P., & Hartel, P. (2003). Service discovery at home. In *Proc. of IEEE Information, Communications and Signal Processing*, pages 1929–1933.

Swanson, B. & Gilder, G. (2008). Estimating the exaflood: The impact of video and rich media on the internet, a "zettabyte" by 2015. Technical report, Discovery Institute, Seattle, Washington, USA.

Teger, S. & Waks, D. J. (2002). End-user perspective on home networking. *IEEE Communications Magazine*, pages 114–119.

Thorne, D. & Bitzer, W. (2010). Home gateway and home network diagnostics requirements. Technical report, Home Gateway Initiative Draft 0.4.

TTCS (2010). Jj-300.00: Home-network topology identifying protocol (HTIP). Technical report, Telecommunication Technology Committee Standard.

UPnP.org (2008). UPnP device architecture 1.1. Technical report, UPnP Forum, http://www.upnp.org.

van Hartskamp, M., kyoung Song, A. Y., Bhagwat, A., Fairman, B., Hlasny, D., Frost, G., Manbeck, J., McQueen, J., Gadiraju, N., Bopardikar, R., fig:overheads Bardini, Chen, R., & Palm, S. (2006). UPnP QoS architecture v 2.0. Technical report, UPnP forum.

van Mieghem, P. (2006). *Performance analysis of communications networks and systems*. Cambridge University Press.

Wei, W., Wang, B., Zhang, C., Kurose, J., & Towsley, D. (2008). Classification of access network types: Ethernet, wireless LAN, ADSL, cable modem or dialup? *Computer Network*, **52**, 3205–3217.

Wireshark.org (2010). Wireshark.

Xbmc.org (2010). XBMC, http://www.xbmc.org/.

Yamazaki, H., Zhang, Z., & Yoda, I. (2005). Resource management technologies for home network services. *Journal on NTT Technical Review*, **3**(1), 12–16.

Yuan, G., Grammenos, R. C., Yang, Y., & Wang, W. (2010). Performance analysis of selective opportunistic spectrum access with traffic prediction. *IEEE Transactions on Vehicular Technology*, **59**(4), 1949–1959.

Zahariadis, T., Pramataris, K., & Zervos, N. (2002). A comparison of competing broadband in-home technologies. *Journal Electronics and Communication Engineering*, **14**(4), 133–142.

Zhu, F., Mutka, M., & Ni, L. M. (2005). Service discovery in pervasive computing environments. *IEEE on Pervasive Computing*, **4**(4), 81–90.

Samenvatting

Thuisnetwerken worden steeds dynamischer en technologisch heterogener. Ze bevatten een toenemend aantal apparaten die veel verschillende functionaliteiten bieden en voor veel verschillende diensten kunnen worden gebruikt. Deze apparaten zijn met elkaar verbonden met behulp van verschillende netwerktechnologieën (bijvoorbeeld CAT-5, draadloze, coax kabel, of elektriciteitskabel). Echter, het onderling verbinden van deze apparaten is vaak niet eenvoudig. De toenemende technologische heterogeniteit heeft het beheren van apparaten en diensten erg complex gemaakt. Daarnaast voorzien thuisnetwerken in de kritische "laatste meters" van de publieke telecom- en Internet-infrastructuur en hebben ze dus een grote impact op de "end-to-end" betrouwbaarheid en kwaliteit van de diensten die deze infrastructuren leveren. Dit biedt dienstverleners de nodige uitdagingen, niet alleen om een goede kwaliteit van de dienstverlening in dergelijk heterogene thuisnetwerken te kunnen handhaven, maar ook om op afstand problemen in het huisnetwerk te kunnen monitoren en oplossen. Dit proefschrift beschrijft onderzoek en verscheidene oplossingen op het gebied van het monitoren van diensten in thuisnetwerken, en dan vooral in de volgende drie gebieden: (1) het automatisch kunnen ontdekken en configureren van apparaten en diensten, (2) beheer op afstand, en (3) het bieden van kwaliteitsgaranties (Quality-of-service, oftewel QoS).

Met betrekking tot het eerste gebied is bekend dat de huidige technologie voor het automatisch ontdekken van diensten in een netwerk is ontworpen om de toenemende rol van de eindgebruiker in het beheren van netwerken en diensten te verlichten. Echter, het ontbreekt de relevante Service Discovery Protocollen (SDP's) aan twee cruciale functies, te weten: (1) ze zijn niet platform- en netwerk-onafhankelijk, en (2) ze beschikken niet over voldoende mechanismen om de bronnen van apparaten (geheugen, processortijd, enz.) te kunnen reserveren. Bijgevolg kunnen apparaten met verschillende SDP's niet met elkaar communiceren en ook geen functionaliteiten en bronnen met elkaar delen op een gecontroleerde manier. Dat komt vooral voor bij apparaten die verschillende netwerktechnologieën gebruiken. Als oplossing voor het eerste probleem hebben wij een nieuwe proxy-server architectuur on-

twikkeld die IP-gebaseerde apparatuur en diensten kan laten ontdekken op niet-IP gebaseerd netwerken en vice versa. We hebben deze proxy-architectuur geïmplementeerd met UPnP respectievelijk Bluetooth SDP als IP-en niet-IP-gebaseerde SDP's. De proxy zorgt ervoor dat Bluetooth-apparaten en UPnP apparaten (de zg. UPnP control points) elkaar kunnen ontdekken, elkaar toegang kunnen verlenen, en elkaars diensten kunnen gebruiken. Uit validatie experimenten met het prototype bleek dat hiermee inderdaad naadloze interoperabiliteit kan worden bereikt waarbij alle benodigde proxy functionaliteiten zich op een enkel apparaat bevinden. Daarmee vereist deze oplossing dus geen verdere aanpassingen van thans bestaande UPnP- en Bluetooth-apparatuur. Hoewel de proxy zelf ook een negatieve impact heeft op de end-to-end prestaties van de te ondersteunen dienst, hebben we aangetoond dat deze prestaties ook met proxy nog wel aanvaardbaar zijn voor een eindgebruiker. SDP's kunnen ook een rol spelen bij het bestrijden van conflicten met betrekking tot het gebruik van bronnen. Daartoe hebben wij een generiek mechanisme ontwikkeld voor het reserveren van die bronnen, met eigenschappen die zijn afgeleid van de huidige werking van SDP's. Experimenten met een prototype tonen aan dat dit mechanisme inderdaad de schaalbaarheid en duurzaamheid van SDP diensten verbetert, zonder veel extra's te eisen van de betrokken processoren.

Met betrekking tot het tweede gebied is bekend dat de end-to-end kwaliteit van Internetdiensten sterk afhangt van de prestaties van het thuisnetwerk. Bijgevolg vereisen dienstverleners de mogelijkheid om betrokken apparaten in het thuisnetwerk, dus achter de home gateway (HG), te kunnen monitoren en configureren. Ze kunnen echter maar beperkt nieuwe eisen stellen aan deze apparaten, die veelal via de detailhandel worden verkocht. De markt voor consumentenelektronica bevindt zich grotendeels buiten het beheersbare domein van dienstverleners. De kansen liggen daarom in het intelligent gebruik maken van de beschikbare protocollen voor controle en beheer van deze apparaten. In dit proefschrift stellen we een architectuur voor die het dienstverleners mogelijk maakt om op afstand apparaten te kunnen ontdekken en beheren in een zeer heterogeen thuisnetwerk. We hebben een proof-of-concept ontwikkeld voor het beheer op afstand van UPnP-apparaten in huis gebruik makend van een TR-069/UPnP proxy op de HG. Hoewel deze architectuur protocol-specifiek is, kan het gemakkelijk worden aangepast aan andere web-services gebaseerde protocollen. Dienstverleners vragen tegenwoordig ook om diagnostische instrumenten waarmee ze op afstand thuisnetwerken kunnen monitoren. Een van deze instrumenten wordt geacht in staat te zijn om informatie over de topologie van het thuisnetwerk te verkrijgen. Hoewel er al zogenaamde topology discovery protocollen bestaan, is er nog niets bekend over hun prestaties. We hebben daarom een set van kritieke prestatie-indicatoren (KPI's) opgesteld met betrekking tot topology discovery in thuisnetwerken en we laten zien hoe deze KPI's moeten worden gemeten. We hebben dit toegepast op het Link-Layer Topology Discovery protocol en het Link-Layer Discovery

Protocol. De meetresultaten tonen aan dat geen van deze protocollen aan alle eisen zoals gesteld door de dienstverleners voldoen.

Met betrekking tot het derde gebied is bekend dat de huidige QoS-oplossingen vooral werken op basis van classificatie van het verkeer. Omdat ze moeten worden ondersteund door alle apparaten in het netwerk, zijn ze relatief duur voor thuisnetwerken. Bovendien zijn de oplossingen behorende bij verschillende netwerktechnologieën niet interoperabel met elkaar. Alternatieve QoS technieken vereisen dat de diensten voor de eindgebruiker pragmatisch hun eigenschappen aanpassen aan de werkelijke toestand van het netwerk op een gegeven tijdstip. Voor deze technieken moet de toestand van het thuisnetwerk in termen van de beschikbare bandbreedte, vertraging, jitter, enz., dus bekend zijn in real time. Voor het bepalen van deze toestand bestaan nog geen geschikte instrumenten. In dit proefschrift beschrijven we een nieuwe methode om de capaciteit en de beschikbare bandbreedte tussen een server en een cliënt in een thuisnetwerk te bepalen. De belangrijkste kenmerken van deze methode zijn: (a) bestaande apparaten hoeven hiervoor niet aangepast te worden, (b) het vereist geen voorkennis van de netwerktopologie op de link-laag, en (c) de meetresultaten zijn nauwkeurig genoeg om betrouwbare QoS voorspellingen te doen voor de meest relevante toepassingen. Om deze voorspellingen effectief te kunnen gebruiken voor het al of niet toelaten en aanpassen van de dienst of content, moet men ook weten hoe de toestand van het thuisnetwerk zich naar verwachting zal wijzigen direct nadat de huidige stand is bepaald. Daarvoor is kennis nodig met betrekking tot de stochastische eigenschappen van het verkeer in thuisnetwerken, maar daarover was tot op heden nog niet veel bekend. Op basis van een relatief kleine aantal observaties in diverse thuisnetwerken in Nederland zijn we echter in staat geweest om een eerste model te bouwen die de dynamiek van verkeer in thuisnetwerken beschrijft.

Acknowledgments

One who seeks the unseekable cannot subsequently be accused of negligence in seeking what is seekable (the Proof of Islam, Imam Al-Ghazali).

This thesis represents not only the end of my journey in obtaining my PhD, it is a milestone of my research career. The project is a collaboration between the Eindhoven University of Technology (TU/e) and TNO. The present thesis could not have been realized without direct and indirect contributions of many people.

First, I wish to sincere thank my supervisor at TNO and copromotor, dr.ir. Frank T.H. den Hartog for the great opportunities he provided by confiding this project to me. His extensive discussions around my works and detailed explorations in each topic have provided a good basis for the thesis. I deeply appreciate his patience, efforts and creativity in guiding me.

I would like to express my gratitude to prof.ir. A.M.J. (Ton) Koonen, my first promotor, for allowing me to pursue the collaboration project between TU/e and TNO. I am truly thankful for him to be the one person that I could always reliably count on to give me support during difficult times. His kind encouragements often gave me motivational inspiration back onto the path pursuing my goal.

I am grateful to prof.dr. Antonio Liotta, my second promotor, for his critical thought about the work and for shaping my manuscripts into a more readable form. Throughout my thesis-writing period, he provides encouragement, sound advice and a lot of good ideas.

I wish to thank my PhD committee: prof.dr. Dave Marples MEng, prof.dr. Johan J. Lukkien, prof.dr. Ernst W. Biersack and prof.dr.ir. Erik R. Fledderus for their constructive criticism and excellent advice during the preparation of this thesis. I am indebted to my friends and colleagues at TU/e: especially Rian, prof.dr.-ing. Leon M.F. Kaufmann, Eduard, Bas, Johan, Desiree, and Nandra; and at TNO: especially Erik, Floris, Bing, Kamal, Ben, Shuang, Martijn, and Maria - for all the help, emotional support, comraderie, entertainment, and friendship they provided. This pertains also to Jose and Marc, who have assisted me in many different ways. Without their interest and cooperation this thesis would not have been possible.

My special esteem goes for prof.drs.dr. Leon J.M. Rothkrantz and his wife, Fien, for their generous help and abiding encouragement when my family faced difficult moments. I also wish to extend my acknowledgement to my colleagues at KPN: especially Marco, Fred, Marcon, Henk, Rein, and Marcel; and to Indonesian families in the Netherlands: especially teh Alia & Csaba, teh Rita & mas Umar, ibu Rosita, family Saraian, family Raams, family Santosa, family Djaya, family Frediansyah, family Tri Hartanto, family Kamarza, family Yudhi, and family Budiarto - for their kind support and encouragement.

My infinite thanks go to my family for their love, continual prayers and encouragement: family Sudjali (bapak, ibu, mbak Renny, mas Ari, Ria, Rica, in-laws, nieces and nephews), family Sabikun (ayah, ibu, and in-laws), and family Geldof. Most of all, I am sincerely grateful to my dearest wife, Siska Fitrianie, for her unyielding and unconditionally loving support, patience and respect. I have been fortunate to have her advice and help as a scientist; this achievement truly is hers as well. Of course, I must not forget about Lavitanea L. Delphinanto, our daughter. No words can express the blessing of having her, except: "You are the source of my happiness, the reason of my thankful (to Allah) and the purpose for me to strive hard".

Finaly, اَلْحَمْدُ لِلَّهِ رَبِّ ٱلْعٰلَمِينَ - *all the praises and thanks be to Allah, the Lord of mankind and all that exists.*

Archi Delphinanto
Delft, 16th January 2012

Curriculum Vitae

Archi Delphinanto was born in Malang, Indonesia, in 1975. He received his B.Sc. in engineering physics from Bandung Institute of Technology, Indonesia, in 1998. Archi started his professional career as a consultant in various software houses in Indonesia from 1998 to 2000. In early 2001, after a homologation Archi was admitted by Stan Ackerman Institute, Eindhoven University of Technology, for pursuing a post-graduate program in Information and Communication Technology (ICT). After completing the program in 2003 he received the degree of Professional Doctorate in Engineering (PD.Eng), and continued to work as a researcher in two Dutch national research projects, namely Residential Gateway Environment (2003-2004) and Feeband B@Home (2004 to 2006).

In 2006, Archi started working on his PhD work with funding from several projects, namely Freeband B@Home, the EU project BONE, and the EU project FIGARO. In 2010 and 2011 he finished this work as a scientist at TNO. During this period, Archi also acted as member of the Team Program Committee (TPC) of several IEEE conferences. In addition to journal and conference publications, his PhD research has resulted in two international patents, two contributions to the Home Gateway Initiative (HGI), and an industrial award at Connected Home Global Summit (London 2011) as the best innovation in "Software Modularity and Applications for Home Gateways". Since October 2011, Archi has joined the Dutch telecom provider, KPN, as a network architect. His main responsibility is to perform innovations on the existing KPN backbone and local networks.

Publications

Conference Proceedings

Mekenkamp, G., Noorbergen, A., Cramer, E., Hillen, B. A. G. , Delphinanto, A. & den Hartog, F. T. H. (2013). Internal Federation of IP and Non-IP Home Networking Middleware. In preparation for *the IEEE Consumer Communication and Networking Conference (CCNC 2013)*, Las Vegas, USA, January.

Castellanos, E. G. D., Delphinanto, A., & den Hartog, F. (2012). Performance analysis of home network topology discovery protocols. In *Proc. of the IEEE Consumer Communication and Networking Conference (CCNC 2012)*, Las Vegas, NV, USA.

Delphinanto, A., Koonen, A. M. J., Zhang, S., & den Hartog, F. T. H. (2011). Real-time probing of end-to-end capacity and available bandwidth in heterogeneous local networks. In *Proc. of the IEEE Consumer Communication and Networking Conference (CCNC 2011)*, pages 828-829, Las Vegas, NV, USA. IEEE Press.

Delphinanto, A., Koonen, A. M. J., & den Hartog, F. T. H. (2011). End-to-end available bandwidth probing in heterogeneous IP home networks. In *Proc. of the IEEE Consumer Communication and Networking Conference (CCNC 2011)*, pages 431-435, Las Vegas, NV, USA. IEEE Press.

Delphinanto, A., Koonen, A. M. J., Zhang, S., & den Hartog, F. T. H. (2010). Path capacity estimation in heterogeneous, best-effort, small-scale IP networks. In *Proc. of the IEEE Conference on Local Computer Networks (LCN 2010)*, pages 1076-1083, Denver, CO, USA. IEEE Press.

Delphinanto, A., Hillen, B. A. G., Passchier, I., van Schoonhoven, B. H. A., & den Hartog, F. (2009). Remote discovery and management of end-user devices in heterogeneous private networks. In *Proc. of the IEEE Consumer Communications and Networking Conference (CCNC 2009),* pages 1-5, Las Vegas, NV, USA. IEEE Press.

Delphinanto, A., den Hartog, F. T. H., & Koonen, A. M. J. (2008). Improving quality of experience by adding device resource reservation to service discovery protocols. In *Proc. of the IEEE International Conference on Communications (ICC 2008)*, pages 1813-1818, Beijing, China. IEEE Press.

Delphinanto, A., Lukkien, J. J., Koonen, A. M. J., Madureira, A., Niemegeers, I. G. G. M., den Hartog, F. T. H., & Selgert, F. (2007). Architecture of a bidirectional Bluetooth-UPnP proxy. In *Proc. of the IEEE Consumer Communications and Networking Conference (CCNC 2007)*, pages 34-38, Las Vegas, NV, USA. IEEE Press.

Delphinanto, A., Koonen, A. M. J., Peeters, M. E., & den Hartog, F. T. H. (2007). Proxying UPnP service discovery and access to a non-IP Bluetooth network on a mobile phone. In *Proc. of the IEEE Symposium of Communications and Vehicular Technology in the Benelux (SCVT 2007)*, pages 1-5, Delft, the Netherlands. IEEE Press.

Journal

Delphinanto, A., Zhang, S., Koonen, A. M. J., Nicolai, F. P., & den Hartog, F. T. H. (2012). Active probing of end-to-end path capacity in highly heterogeneous private networks. *IEEE Transactions on Network and Service Management*. Submitted.

Delphinanto, A., Castelanos, E. D., Liu, B., Liotta, A., Koonen, A. M. J., & den Hartog, F. T. H. (2012). Traffic prediction in home network. *Journal of Network and Systems Management*. Springer. Submitted.

Delphinanto, A., Koonen, A. M. J., & den Hartog, F. T. H. (2011). Real-time probing of available bandwidth in home networks. *IEEE Communication Magazine*, **49**, 6, 134-140.

Patent

Delphinanto, A., Nicolai, F., & den Hartog, F. T. H. (2011). *Network transmission capacity measurement, the Netherlands, WIPO Patent WO/2011/008090*. 10 January. Granted.

Delphinanto, A., Madureira, A., & den Hartog, F. T. H. (2009). *Personal networks proxy-bridge for connecting different types of devices, U.S. 20090303926*, United States Patent. 10 December. Granted.

Contributions to Home Gateway Initiative

den Hartog, F. T. H. & Delphinanto, A. (2011). *UDP throughput and packet loss measurements of HomePlug.* Contribution HGI01865 Home Gateway Initiative (HGI). 16 October.

den Hartog, F. T. H. & Delphinanto, A. (2010). *A renewed plea for active home network probing.* Contribution HGI01452 to Home Gateway Initiative (HGI). 25 March.

Award

Delphinanto, A. & den Hartog, F. T. H. (2011). *Best innovation in software modularity and applications for home gateways.* In Connected Home Global Summit 2011. London, UK. May.