Personal Networks
An Architecture for Self-Organized Personal Wireless Communications

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 17 juni 2008 om 10.00 uur

door

Martin Edvard JACOBSSON

Magister i Datavetenskap van Linköpings Universitet, Zweden
geboren te Boo, Zweden.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. I.G.M.M. Niemegeers

Samenstelling promotiecommissie:

Rector Magnificus,	Voorzitter
Prof.dr.ir. I.G.M.M. Niemegeers,	Technische Universiteit Delft, promotor
Prof.dr.ir. S.M. Heemstra de Groot	Technische Universiteit Delft
Prof.dr.ir. L.P. Ligthart	Technische Universiteit Delft
Prof.dr.ir. H.J. Sips,	Technische Universiteit Delft
Prof.dr.ir. E.R. Fledderus,	Technische Universiteit Eindhoven
Prof.dr.ir. I. Moerman,	Universiteit Gent
Prof.dr. R. Prasad,	Aalborg Universitet

ISBN 978-90-9023196-9

Keywords: Wireless communication, Ad hoc networking, Self-organization

This research was partially supported by the European Commission under the research projects IST NEXWAY, IST MAGNET, and IST MAGNET Beyond and the Dutch Ministry of Economic Affairs through the project PNP2008 under the Freeband Communication Impulse of the technology programme.

Printed in The Netherlands

Acknowledgements

This thesis describes work that I performed at Wireless and Mobile Communications group at Delft University of Technology under Prof. Ignas Niemegeers. It was his thoughtful guidance that gave me the opportunity to understand the research process and the scientific community, first during my M.Sc. thesis work and then during my Ph.D. thesis work. His encouragements, enthusiasm, support, and insights have been invaluable. Further, I would like to thank the committee members for spending time and providing feedback to this thesis.

Thanks also to all my colleagues with whom I have had many fruitful and/or joyful discussions. In relation to this thesis, I am especially grateful for the cooperation with Cheng Guo, Jinglong Zhou, Venkatesha Prasad, Anthony Lo, Novi Ineke CempakaWangi, Weidong Lu, and Ertan Onur. I am also grateful to Przemyslaw Pawelczak and Bao Linh Dang for for helping me out taking the photos for this thesis. In addition, there are also a couple of M.Sc. students who contributed to this work by doing their final projects with me. I would like to acknowledge them here: Yan Gao, Xiang Han, Cheng Guo (again), and Ting Liu.

Thanks to IST MAGNET, MAGNET Beyond, Freeband PNP2008, IST NEXWAY, and IOP GenCom QoS for PN@Home, I had the privilege to work with many excellent researchers. They provided me with plenty of opportunities to discuss and enhance the concepts and ideas that lie behind this thesis. By getting to know you, I am sure that we will find ways to mutually enhance each other's work also in the future!

Finally, I want to thank my family. This is of course not only limited to my parents, my brother, and my wife, but also uncles, aunts, cousins, and all my Indonesian and Dutch in-laws. Special thanks to my brother-in-law for translating the summary and propositions to Dutch. Without my wife, I would never have become a PhD student in Delft!

Tack, Terima Kasih, Bedankt, Thank You!

Martin Jacobsson
Delft

To My Wife.

Contents

Chapter 1

Introduction

Since the dawn of time, communication has been an integral part of human life and the need of better technology to support our communication has never ceased to exist. Over the centuries, we have invented many different ways of communication to bridge the barrier of both distance and time. With people becoming increasingly nomadic these days, the need for communication with business partners all over the world and with loved ones while on the move have never been more pronounced, which is something the world wide success of mobile telephony has demonstrated. To this end, technology has been developed to ease communication while being mobile. Migrant workers overseas may easily, for a relatively small fee, have voice conversations with family back home on the other side of the planet. At the same time, the mode of communication becomes more varied and richer. Looking back not too long time ago, only primitive forms of communication could bridge any significant distance, such as hand-written letters, smoke signals, and Morse-coded telegraph messages. Today, nothing stops us from sending video and audio messages to any place on earth.

1.1 Past, Present, and Future Telecommunication

Telecommunication technologies, both wired and wireless, are what makes rich communication while on the move possible. Furthermore, the merger of telecommunication and computing is the enabling factor for rich communication. However, this does not stop with human interaction. Technology is being used and can be used to automate many tasks. For example, with home automation, we can control every little electronic device in our homes. With electronic agendas accessible from everywhere, we can better plan our daily activities. By using sophisticated entertainment devices, we can listen to music, watch movies, or play games while waiting at the bus stop or at the airport. Whatever possibilities the future holds, one can only guess.

The Internet started in 1969 as a research project and grew into a world wide network in the second half of 1990s, connecting computers all over the world. Popular services such as e-mail, World Wide Web and peer-to-peer file sharing evolved and made the Internet attractive also for the common man. The growth of Internet has been remarkable and has reached 60 % of the population in the western world [93]. But, the growth does not stop there. While the growth of Internet penetration is slowing down, the achievable data rates continue to increase and this will enable new services to the users. Soon, it will be possible to broadcast television and video on demand over the Internet to everyone everywhere.

Mobile telephony is yet another example of a very successful technology [51]. The first successful mass market deployment of mobile telephone systems started in the 1980s. In less than twenty years, the mobile phone has gone from being a rare and expensive device accessible only to business men with an interest in high-tech gadgets to a pervasive low-cost personal item for everybody. In many countries, mobile phones now outnumber landline telephones, with most adults and many children now owning mobile phones. In 2005, there were 2.17 billion mobile subscribers worldwide but only 1.26 billion landline subscribers [234]. While Global System for Mobile Communication (GSM) is currently the leading mobile technology standard, others, such as Universal Mobile Telecommunications System (UMTS) and the UMTS extension High-Speed Downlink Packet Access (HSDPA), will soon take over. These technologies offer better packet switching support as well as higher data rates with similar support for mobility. Another recent promising technology that can bring high data rates to the mobile user is IEEE 802.16 [88][89], also known as WiMAX. With these technologies we can soon watch movies while on the move. However, this is probably just the start of the hunt for higher data rates for mobile devices. Better battery technology (or other miniaturized energy sources), more computational power, and improved radio technology will undoubtedly offer better data rates and more communication possibilities.

While Internet and mobile telephony have been developed side by side, there are also attempts to integrate the two. Nowadays, there are plenty of websites on the Internet where one can send Short Message Service (SMS) or Multimedia Messaging Service (MMS) messages to mobile phones. Conversely, we have mobile phones that can send e-mails and while it is possible to connect directly to the Internet from a mobile phone, this is still not very wide spread. Beyond any doubts, this trend will definitely continue as normal users do not wish to have two separate networks; one when on the move and another one when at home. Instead, users expect the two networks to be the same and fully integrated.

The evolution of radio communication has also given birth to another direction; short range wireless communication. One of the first successful mass market product in this segment was the Wireless Local Area Network

(WLAN) standard IEEE 802.11 [82] originally released in 1997. It was designed to make the LAN wires redundant in an office and was much more successful in this than any of its predecessors such as the Infrared Data Association (IrDA) [92]. When the enhanced version IEEE 802.11b came, the deployment really took off. So called hotspots were installed where an IEEE 802.11b (and later IEEE 802.11g) access point could offer wireless Internet connectivity with data rates of several Mbps to devices such as laptops and Personal Digital Assistants (PDAs) within a range of up to about 100 meters. Millions of hotspots have been installed world wide in strategic locations where people congregate and/or need to communicate, such as airports, train stations, cafés, hotels, and convention centers. Nevertheless, this technology will never be able to achieve the same coverage as GSM and UMTS.

To connect wearable and hand held devices around a person, a range of 10 meters is enough. This has led to the development of yet another branch of technologies that have very high data transmission rates, low power consumption, but limited range. They usually go under the term Wireless Personal Area Networks (WPANs) by which IEEE 802.15.1 [84] (commonly known as Bluetooth) is currently the most common WPAN technology. These technologies promise to connect mobile phones, laptops, PDAs, and other personal devices located within 10 meters in a seamless way with high data rates and low enough power consumption for normal battery-powered devices. Typical WPAN communication takes place between a person's mobile devices, such as a camera requesting time and location information from a Global Positioning System (GPS) receiver to tag a picture or a mobile phone sending voice to a wireless headphone. It can also be information sharing between two persons meeting on the street. For instance, they can share recently taken pictures or interesting locations (geographical data) that one of them just has visited. Even in this segment, higher data rate versions are to be expected in a near future, such as the IEEE 802.15.3 family [85][86].

The current research and development of wireless communication brings us more specialized communication technologies that focus on a particular communication problem and thereby can better address their particular niche problem. Figure 1.1 shows the current landscape of wireless communication technologies. It shows how each wireless communication technology targets a specific area of need as there currently is no ultimate wireless communication technology available that can offer high data rates, long range, and, at the same, time low power consumption. It seems that we need to cope with several quite different wireless technologies for many years to come. The down side of this development is that we also get a multitude of various different communication techniques and protocols and currently there is a clear lack of integration between them. This, in turn, leads to problems for the end-user that has to understand and master all the specialized technologies and accept that they do not fully work together. Therefore, our focus should be on how to make them best complement each other and work together seamlessly.

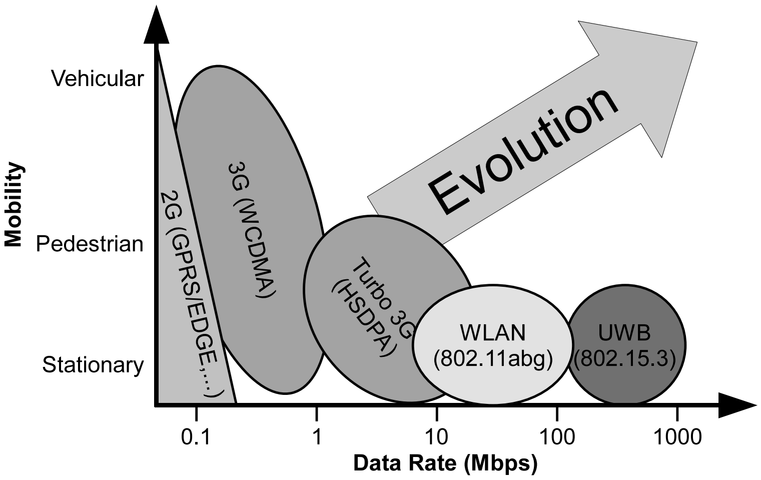


Figure 1.1: Wireless Communication Landscape

Regrettably, very little effort has been made to integrate these different technologies. It is possible to send e-mails from a mobile phone and SMS from an Internet-connected Personal Computer (PC), but the possibilities should go well beyond this. Instead, users are forced to learn each system and manually configure it to inter-operate. In many cases, this is simply not possible because of limitations in the software and this problem just becomes worse as people tend to use more and more electronic devices.

In parallel to this, device technology has made rapid progress in the last decades. Our mobile phones become smart mobile computers and still retain their original form factor. Even the cheaper mobile phones of today can play music, take and view photos and video clips, and even “surf the web”. Micro electronic research has enabled smaller chips that are consuming less energy and yet are more powerful and capable of things one could hardly imagine before. This has lead to a large variety of different devices and terminals. Everything from small and simple mobile phones and music players to PDAs, tablet PCs, and advanced mobile multimedia or entertainment platforms. See Figure 1.2 for some current examples. Hence, there is no reason why future terminals should be a limiting factor for enhanced interoperability.

1.2 Personal Networks

As discussed in the previous section, the shortcomings of current wireless communication technologies are hampering the adoption of sophisticated communication systems by the mass market. The careful reader will notice that all devices in Figure 1.2 have screens. That is because communication is cumbersome and this forces us to interact directly with every single device, using screens and other input and output means. Therefore, new technologies should be centered around the user, improving the quality of life and adapt

Figure 1.2: Examples of Electronic Devices

to the individual, without the need for the user to be aware of the technical details. The devices and the environments need to become smarter, more responsive, and more accommodating to the needs of the individual. Future technologies must aim at integrating these technologies and develop unified and seamless solutions that are easy to use. Further, personalization and ubiquitous access to information and communication will be essential. Any communication system must adapt according to the situation and allow its users to use the most suitable means of communication and to access the most relevant information. As a consequence, new research fields are emerging that aim to provide users with the same service experience independent of user interfaces, terminal capabilities, communication technologies, and network and service providers. Examples are pervasive and ubiquitous computing as well as ambient intelligence and networking.

Personal networks (PN) [163] is a concept related to pervasive computing with a strong user-focused view. While a PAN connects a person's devices around him, a PN extends that PAN with other devices and services farther away. This extension will physically be made via any kind of wired and wireless networks. This can include devices and networks around him/her in the car, office or else where. But, a PN needs to be more than just connectivity. A person's PN must also support the person's applications and take into account the person's context, location and of course communication possibilities. A PN must adapt to changes in the surroundings, be self-configured and support many different types of networks and devices to be as useful and easy to use as possible. Figure 1.3 shows what a PN could look like for a user.

There are many different ways of integrating the various communication technologies and achieving one unified system. The best and the most com-

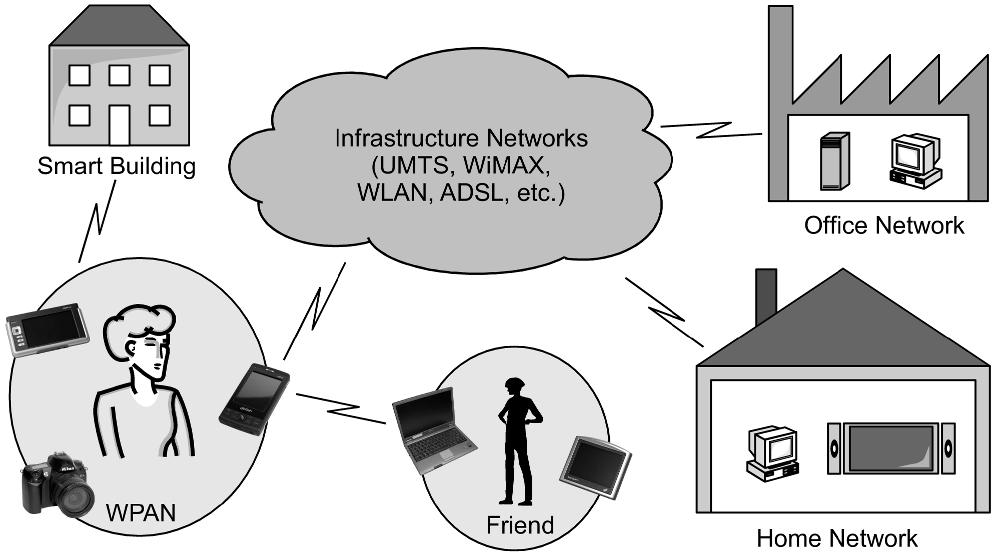


Figure 1.3: The Concept of Personal Networks

plete integration approach is to define a common network layer to be used by all similar to the approach taken by the Internet with the Internet Protocol (IP). Such a general and common network layer architecture that imposes minimal changes to the underlying network types can bridge different communication technologies and offer a homogeneous and clear view to the end-user. At the same time, the network architecture needs to be flexible enough to support all kinds of applications. The key to a successful PN realization is a PN that addresses all a person's communication needs. The PN must include not only the person's wearable and wireless devices but also devices at home, in the car, and in the office. This means that the network layer of the PN must work as a home network at home, a car network in the car, a PAN around a person and glue all these networks together in one PN and at the same time cooperate with existing networks such as the Internet and other fixed networks.

As will be shown later in this chapter, this thesis focuses on the network layer mechanisms of a PN. The success of PN does not only require easier and seamless integration at the network layer, but also new types of interesting and useful applications. At the same time, a PN must be smart, responsive, self-configured, respect device hardware constraints, work well on bad or intermittent connections, and obviously also secure. To better introduce the concept of PN, two user scenarios are given below that demonstrate the possibilities of a PN and what types of applications can benefit from a PN.

1.2.1 Traveling Saleswoman Scenario

One major potential benefit of using PNs is seamless access to resources anywhere. Personal files stored at home or in office can be obtained by

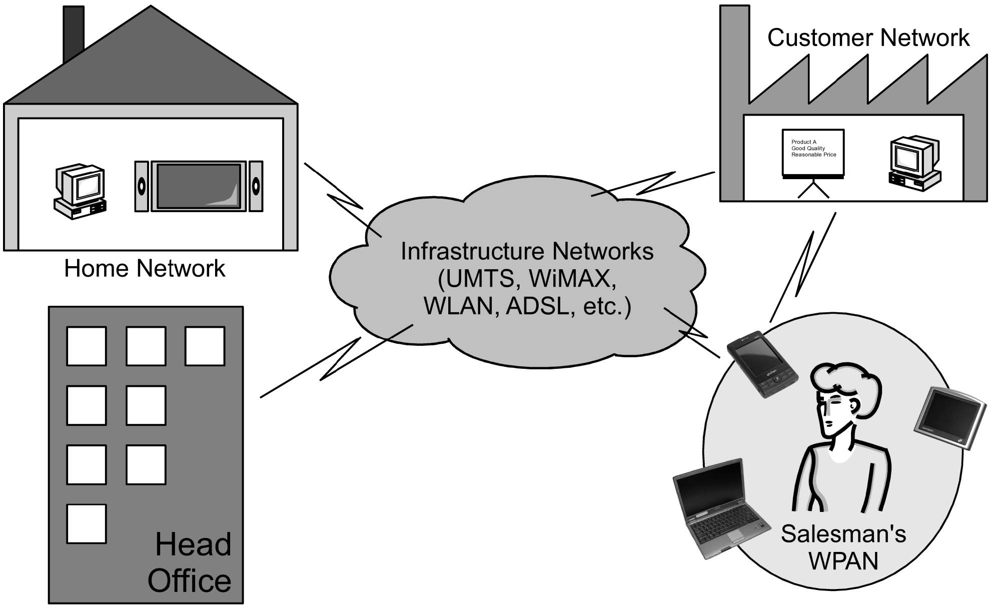


Figure 1.4: Traveling Saleswoman Scenario

one's devices as long as there is some kind of network access. Consider the saleswoman in Figure 1.4, who needs to travel a lot. For such a person, it might be important to always be able to access information and services in the office and to communicate with customers, even when on the move. In addition to this, the person might be equipped with a mobile phone, laptop, headset, etc. and these should be forming a communicating cluster of co-operating devices. The person's PN should offer a framework that enables these devices to seamlessly cooperate and to communicate with distant devices, such as desktop computers, company servers, customer services, home entertainment systems, etc.

With a PN, the saleswoman can access her agenda on any device wherever she is and at the same time make sure her secretary has an up-to-date copy as well. The same holds for personal and shared files. When at a customer, she can share some of these files with the customer in order to be able to present products, make offers, etc. These are very simple applications, yet very important ones. They must work with whatever network access is available. For instance, when at a customer, they should be able to use the customer's network to improve transmission speeds.

Further, there is also a need for non-business-related applications. To be away from home and family for extensive times can sometimes be demanding. Screens, cameras, speakers, and microphones at home can form a second communicating cluster of cooperating devices in the PN. The PN will enable these devices and the devices carried by the saleswoman to communicate and cooperate and thereby offer her the ability to communicate with her family. The devices at home provide her with a virtual home environment through which rich communication can take place. Through this environment, she can virtually see her family, talk with them, and even play games.

Depending on the communication requirements, she could also continue all this while traveling. She could listen to streamed music from the home entertainment system while driving, or play a game while waiting for the airplane, etc. If she meets a friend somewhere, a temporary communication network can be established, to share files, services or just to play a multiplayer game for a while.

While several existing technologies can offer solutions to some parts of this scenario, there is still very little work on combining these technologies into a seamless integrated solution for a normal user. Today, employers have experts that sets up servers and configures wireless devices to inter-operate with their enterprise software on behalf of their employees. Even so, these solutions are typically application-specific and will not work for new applications without proper integration. For the end-user, they are far from seamless. Complex settings cause frustrations and make people wonder whether it will work at the next customer visit. PN tries to address this issue by aiming at being easy to use, setup, configure, and maintain, as well as fast and secure.

1.2.2 Care for the Elderly

PNs can be a powerful tool for personal communication if they are designed to interact with other PNs as well as existing networks and services. With an aging population, this may prove to be a very important function of PNs. An elderly person could be equipped with a PN consisting of various medical sensors to continuously allow monitoring of the health of that person. Such sensors could include blood pressure and heart beat sensors, activity sensors, accelerometers, positioning devices, and more. When something happens, the PN can alarm certain parties about the incident. This can involve care-takers, trusted near relatives, friends, and neighbors. A care-taker or an emergency responder can try to make contact with the elderly through a device he or she is carrying to find out more details about the incident. Otherwise, the location device may inform about the person's location so that medical staff can be sent there immediately. This may allow the elderly person to leave the house, knowing that help is still available even if something goes wrong.

Another requirement for good care of the elderly is good communication, not only in emergency situations. To be able to assist well in daily tasks of an elderly person, care-takers, relatives, friends, neighbors, and of course the elderly need to communicate. This can be about who should do the groceries, assist in cleaning the house, going to and from the doctor and pharmacy. The PN can help the involved parties in coordinating their efforts in a more efficient way. Shared agendas can be established, where each individual's agenda can be used in the planning. Figure 1.5 illustrates this scenario.

Also for the elderly, a PN can be very useful. It could improve the elderly's

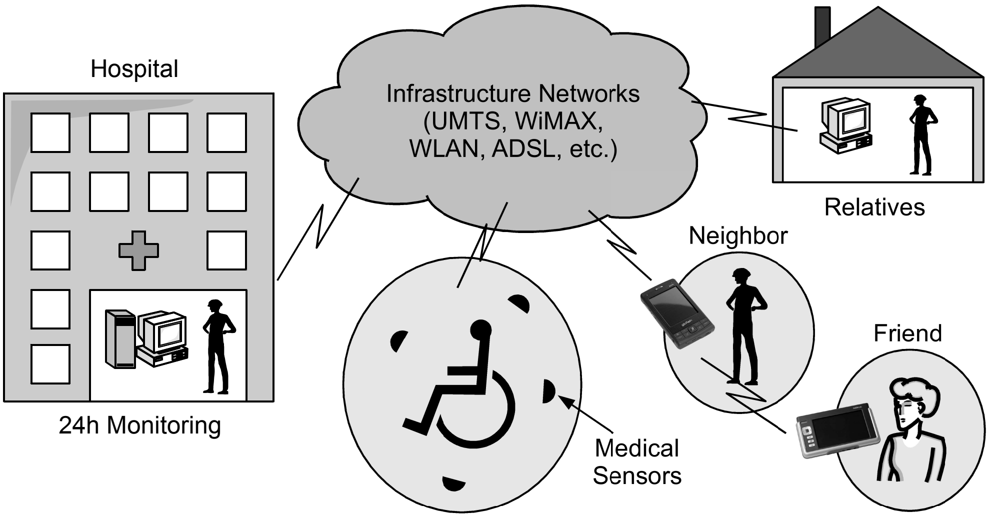


Figure 1.5: Care for an Elderly

capability to communicate with friends, who might also be elderly or it can remind about various things, such as when to take certain medicines for those whose memory is fading. However, designing a PN for the elderly is even more challenging because of an even greater requirement for usability. Such a PN must work for people that may not at all be accustomed to modern electronic devices or have lost their ability to deal with complexities. Further, it must also be usable for people who have reduced audiovisual capabilities and/or movement disorders, such as tremors in arms and hands.

This area of application poses a significant challenge for PNs since it requires ease of use for several very different groups of people, efficient and reliable communication, and also security. The system must be dependable, meaning that you can depend on the system also in emergency situations. Privacy is another complex area that can not be neglected. While the elderly wants fast response in cases of emergency, he/she may not want to be monitored in detail all the time by unscrupulous relatives, neighbors, etc.

The traveling saleswoman scenario and the care for the elderly scenario highlights some of the potential application areas of PNs. More scenarios that reflect the vision of PN have also been defined elsewhere. See, for instance [163][96][132].

1.3 Research Motivations, Targets, and Scope

As can be seen from the endeavor set out in previous sections, a lot of research, design, and development are required before we reach the goal. On top of that, standardization work is required to guarantee the success of seamless integration of the whole range of possible PN devices and networks. In other words; with four years of hard doctoral research, we are only able to scratch the tip of the iceberg. As a consequence, this thesis work had to focus on some very specific aspects.

The focus of this thesis lies in the network layer with only a few other related topics being covered, mainly on service frameworks and cross-layer issues. We will build on existing technologies for wireless communication and perhaps future improved ones outlined earlier in this chapter. Then, an architecture mostly based on network layer concepts is defined. Though the architecture goes well beyond networking, this thesis will focus on the network issues of this architecture. Such as, how can the PN network layer configure and organize itself without requiring user intervention? How can it find communication possibilities and route traffic to the destination? How to protect the networking mechanisms from malicious attackers and how to protect the user's privacy?

In the end, not all networking aspects will be covered. We will not consider the networking issues that may arise at the network providers. It is clear that specialized support systems at the operators could be made for PNs. However, instead we assume that network access remains unchanged which will allow a faster deployment of PNs. Hence, we focus on a PN that is as independent as possible from both the wireless technologies and the network access technologies.

1.4 Research Methodology

The research approach we took started from formulating requirements, analyzing the state of the art, and identifying the gaps. Then, we defined a high level architecture for a potential solution. At this early stage, we did our utmost to validate its usefulness and feasibility. Subsequently, we started the work to make the architecture concrete by looking at solutions for the various architectural components. For each major component, we proposed a solution, studied it with simulations, measurements, and finally implemented a full-fledged prototype. The various component prototypes were combined into one more complete prototype of a PN. Basic functionality was tested and thereby we verified the feasibility of our proposed solution. Future research will further validate its true usefulness.

1.5 Thesis Overview

This thesis is organized as follows. In Chapter 2, we present the user requirements we deem necessary for a PN followed by work related to PNs and analysis thereof. Chapter 3 introduces an architecture for PNs capable of fulfilling these user requirements. The architecture functions as a platform where each of the subsequent chapters addresses a particular aspect of the architecture.

In Chapter 4 to 8, the details of the architecture are explained and further worked out. Chapter 4 addresses the formation of what is called Clusters; personal devices that can directly connect to each other with PAN technologies, i.e., how the personal devices can find each other and establish secure communication among themselves. Chapter 5 addresses a particular network aspect of Clusters, namely broadcasting. Several techniques are investigated to efficiently flood messages through out the Cluster by means of relaying devices. Chapter 6 introduces link quality assessment and routing using cross-layer information. In Chapter 7, we focus on forming and maintaining a PN by interconnecting the dispersed personal Clusters by inter-Cluster tunnels over the infrastructure. Chapter 8 looks at how PNs can communicate with other PNs or non-PN devices.

This thesis is finally rounded up by conclusions and future perspectives in Chapter 9.

1.6 Contributions of this Thesis

Most research in this thesis has been done in a collaborative way in various research projects, including IST NEXWAY, IST MAGNET, IST MAGNET Beyond, Freeband PNP2008, and IOP GenCom QoS for PN@Home. The purpose of this section is to highlight the contributions of the author, list the supporting publications, and acknowledge people that have contributed.

The related work of Chapter 2 has been compiled in its entirety by us. The requirements are a re-work and improvement based on earlier work by Weidong Lu, Ignas Niemegeers, and Sonia Heemstra de Groot. Supporting publications include [97].

We were the first to propose an architecture for personal networks based on device ownership, which is presented in Chapter 3. Collaborative work within IST MAGNET certainly enhanced the architecture in several ways, but the main idea still remains. The architectural discussions is our work, but obviously based on discussions within the projects and elsewhere. Supporting publications include [96], [133], and [181].

The Cluster formation of Chapter 4 is a direct consequence of the PN architecture and hence was part of our initial PN architectural specification. Further, the work on anonymity aspects, trust relationships on the link layer, and the securing of broadcast traffic are all our work. The prototype was also completely developed by us. Supporting publications include [135].

The flooding work of Chapter 5 is a contribution by us based on the MSc thesis works of Cheng Guo and Ting Liu, whom we supervised. The protocol design was done in close cooperation with Cheng Guo. The experiments were done by us. Supporting publications include [99].

The work in Chapter 6 was done in collaboration with Jinglong Zhou. While he developed the software and carried out the experiments, we both discussed and contributed to the directions taken in the task regarding the link quality assessments. The work on packet size influence and end-to-end quality is our contribution. Supporting publications include [241], [242], and [243].

The basics of inter-Cluster communication of Chapter 7 was specified by us as part of the initial PN architecture. Many of the details originate from discussions in IST MAGNET and Freeband PNP2008, such as the infrastructure support. We must also acknowledge the work on the latter by Venkatesha Prasad. The prototype was completely developed by us. Supporting publications include [136], [186], [123], and [185].

The work on foreign communication in Chapter 8 was done by us with the support of Venkatesha Prasad. Supporting publications include [100], [139], and [75].

Chapter 2

Requirements and Related Work

Before we start developing new solutions for personal networks, it is important to understand what exactly needs to be solved and what has already been solved. In Section 2.1, we will more clearly formulate what still need to be solved as a small set of high level user requirements and argue why they are important. In Section 2.2, we will look at some earlier and current visions and approaches similar to personal networks. We briefly investigate the related work in conjunction with the user requirements in Section 2.3. From this, it is clear that more research and development is needed. We need to build on existing technologies and then fill the remaining gaps. Section 2.4 summarizes this chapter.

2.1 Requirements for Personal Networks

It should be clear that in order to realize the concept of personal networks, new network solutions are required that can accommodate personal services and applications over a dynamic communication environment. To better understand what needs to be achieved, we list a set of important user requirements that need to be solved. The requirements listed here are evolved versions of [163][181]. These requirements should be seen as additional requirements to what has already been solved by previous work. At the same time, together they capture the total vision of personal networks, which also means that some requirements go beyond the scope of this thesis.

The idea is that later on, we can validate the solutions proposed to materialize the vision by verifying them against these requirements. If all requirements have been addressed to a satisfactory level, we have reached the target; otherwise, additional solutions or revised solutions are still required. However, these requirements are described at a very high level and hence, it is impossible to precisely define if a requirement has been fulfilled. It is nevertheless important to try to formulate requirements, at least to a degree that they can direct the research towards relevant issues.

The following subsections contain the user requirements we consider important. They have been grouped into eight categories, each actually containing several requirements. However, we do not aim at identifying every single individual requirement as this is neither possible nor important at this moment.

2.1.1 Ubiquitous networking

Since personal networks is all about communication, this requirement should not come as a surprise. Devices surrounding a user should form a private personal area network (P-PAN) that enables communication using the available wireless communication technologies. Both current and future WPAN technologies should be supported. Furthermore, the connectivity of the P-PAN must be extendable to devices beyond the close vicinity of the user by means of infrastructure-based wireless access networks, such as UMTS networks, WLAN hotspots, WiMAX, etc. Personal networks must be able to use any type of access technology and therefore be as independent as possible from infrastructure. Regardless of what network or device type, communication must be possible between any device belonging to the user whenever there is connectivity at the link layer. Hence, personal networks should support a heterogeneous network environment by integrating all present network types into one ubiquitous network for the user.

From the two scenarios in Section 1.2, we learned that communication with other person's devices and non-personal devices is also crucial. Ubiquitous communication over heterogeneous network environments with others regardless of the geographical location of the devices must therefore be possible as well.

Since many of the devices will be wearable or otherwise mobile, it is absolutely necessary to handle mobility. It can be devices that roam through networks, links that breaks, or new links that become available. The personal network must be aware of these events and have mobility management mechanisms that can adapt so that ongoing communication can be sustained.

All these networking issues need to be supported in a ubiquitous way, meaning that only minimal user intervention is required. All networking mechanisms must happen without the knowledge of the user. The personal network needs to be able to establish and maintain itself on its own. In other words; personal networks must be self-organized.

2.1.2 Respecting heterogeneous hardware constraints

A personal network will consist of a wide range of different mobile and stationary devices, wireless technologies and networks. The personal network must operate efficiently in such a heterogeneous environment by for instance switching communication paths between different devices, links, and networks to achieve the best possible performance even when the number of devices and the amount of traffic is becoming large. Mechanisms must make sure that devices that rely on battery power do not have to carry the heavy burden and that the utilized devices have the required computational power, memory, bandwidth, and other required capabilities to carry out the tasks in a satisfactory way.

It is true that future technologies will bring us yet faster computational power with less energy consumption and smaller devices with a more robust design at a cheaper prize. Further, battery technology is also improving and new alternative energy sources for mobile devices are becoming available. The smallest devices that are believed to become full participants of a personal network are still not the simplest. Sensor devices or similar have so tight hardware constraints that specially designed techniques are required. On the other hand, there is no real need for such simple devices to fully become part of a personal network. We can therefore require somewhat more capabilities of devices that needs to fully participate in a personal network. However, the personal network mechanisms must still run on battery-powered devices and try to extend the life of the battery as much as possible.

The simplest devices that we would consider for personal networks are wireless headsets, wrist watches, and other wearable devices. Currently, all these devices run on batteries that need to be recharged or replaced after some time. Using today's technology, it is acceptable for a device in a personal network to be able to run for one or a few days before needing to be recharged again. Hopefully, developments in low power consuming electronics and improved battery technology will make this unnecessary in the future.

2.1.3 QoS and reliability

Several potential personal network applications have high demands for end-to-end quality of service (QoS), such as interactive applications, voice and video conferencing, etc. The entire system should meet the demands of these and other applications with respect to QoS and reliability. Thus, parameters such as bandwidth, bit error rate, and latency, should be considered in the routing and the mobility management of a personal network. The personal network must be able to select communication paths that meet these expectations. In some cases, it is necessary to make different network technologies cooperate and to properly impose the QoS demands in each of them to fulfill the end-to-end demand of the applications.

The dynamic behavior of mobile wireless systems calls for very efficient adaptability to meet the demands of the user. The mobility management must be very fast to respond to events such as broken wireless links, changes in bit error rates, or malfunctioning devices. If this is not the case, then the QoS requirements will be violated, making personal networks useless for many important applications when devices are mobile.

Another important requirement for personal networks is reliability. Health applications, such as the ones outlined in Section 1.2.2, need to depend on the system and that it works when it is needed the most. Given the unreliability of mobile and wireless systems, this is a major task. Personal networks should therefore not depend on one single network, but use many different networks and technologies at the same time to reduce the risk of being completely without communication possibilities at any given time. The level of reliability required by the applications should dictate how proactive personal networks should be in finding and keeping backup links as this may imply extra power consumption and perhaps cost.

Reliability can also be about instant data access and prevention of data loss within a personal network. Important data should be backed up to enable access to it at any time as well as protecting it from being lost. Otherwise, this may cause permanent loss of important data when devices are lost or break.

2.1.4 Naming and service management

The personal network provides a network architecture for applications and services to build upon in order to provide a complete solution for the user. However, applications and services still require additional software support to be easy to build and better meet all user requirements. Technical aspects of the network mechanisms should be hidden from both the user and the applications. This will make the system more integrated and at the same time it becomes easier to build applications and services.

Techniques to hide irrelevant aspects of the network layer include naming solutions as well as service discovery and management. Naming is needed to hide addresses and other irrelevant details of the network layer. Names can have meanings to the users and gives a human-understandable handle to relevant objects, such as devices, services, resources, and other objects. Naming is therefore very crucial for user-friendly personal networks. The names can be assigned by the user to give an extra level of personal touch and in order to better organize the resources within the personal network. Furthermore, network addresses may change, but names will remain unchanged until the user changes them. It is therefore better to use names to identify various objects.

To achieve as much self-configuration as possible, we make use of the service abstraction. A service is an entity that offers client applications something useful for the client through a known interface. The type and the capabilities of the service are described in a standardized way [193]. These descriptions can be used by a service discovery mechanism to enable the applications to easily find available services and select the most appropriate one. Furthermore, a good management framework is also required that can manage not only the services, but also the clients and their service sessions. Management means controlling the service usage when the network situation changes so that the clients and the services can operate optimally.

2.1.5 Context awareness

Context information is anything that can characterize the situation of an object, such as a person, device, or network [6]. This information is valuable since it can influence the behavior of the personal networks applications or the personal networks themselves. The more information available to an application, the better that application can respond to the user and the situation. While this additional information is not absolutely crucial for a personal network or its applications to operate, it may still be necessary in order to reach the user's high expectations.

Today's users expect their devices and applications to be intelligent, properly predict the users' intentions, and to automatically adapt to a changing environment. It is therefore necessary to implement a context information framework that can discover, process, and distribute relevant context information. Furthermore, the personal network and its applications must be able to properly use this information. That is, both the personal network and its applications must be context aware.

2.1.6 Security and trust

The new characteristics and possibilities offered by systems like personal networks lead to new security and trust problems that need to be addressed properly [211]. Personal networks can only succeed if people trust it, but unfortunately, personal networks are extra vulnerable because of their mobile and wireless nature. In the world of mobile communication, IT security meets traditional security and this opens up a completely new world of problems in the security domain. The ad hoc nature of personal networks means that a person's personal network will encounter many unknown parties but must remain properly protected from the unknown or untrusted parties.

The main challenge to security for personal networks lies not in the security algorithms or security protocols. There is a rich plethora of security solutions on which personal networks can leverage. The problem is rather to formulate a way for personal networks to model trust among persons, devices, and networks that is both powerful enough to contain all the necessary details and at the same time be comprehensible for a normal user. Trust models and their security systems may become too complicated leading to that the users are severely bothered by them or even fail to sufficiently understand them [19]. At the same time, they fail to protect the right things, because it is no longer clear what to protect and against who in a world of mobile and ubiquitous communication [211].

Even so, a security system is needed and it must protect the personal networks and their users from unauthorized usage. It is important to note that the security system must be an integral part of the system design and not a later add on. The security must also work when devices are stolen or compromised. Tamper-proof is too costly and difficult for consumer products such as personal network devices [210]. Hence, something else must maintain the security of a personal network when devices are lost or stolen.

2.1.7 Privacy

As more and more information in our lives is being digitized, privacy is becoming an even bigger issue [62]. Privacy is about protecting the data kept inside the personal network as well as preventing the possibility to track a user's activities through his/her personal network. First, traffic between a user's devices must always be encrypted. However, this is not enough. In the world of mobile communication, there is an increased risk of theft of devices. We already know from experience that a stolen or forgotten laptop or PDA may lead to confidential information coming in the wrong hands. Unfortunately, with personal networks, it becomes even more important because of the extra capabilities a personal network device will have. Hence, we need to build a system that minimizes the impact in case of lost or stolen devices.

Another privacy problem is that being recognizable by the devices you carry can also be an intrusion to a person's privacy, since this information can be used to track a person's movements and activities. This can be done since many wireless devices expose their identities in forms of link layer addresses or other unique and fixed identities. Anonymity is therefore needed in personal networks. In terms of wireless networks, it means that a device must never expose its identity or anything that can easily be linked to its identity to non-trusted parties [200]. This is important since we are likely to carry the same devices all the time and this can be used by unauthorized persons to track the movements and activities of personal network users. However, preventing all types of identity exposure is an almost impossible task. What we can do, is avoiding the most obvious pitfalls, such as transmitting fixed link layer addresses. If that is done, more sophisticated methods are required in order to track someone.

2.1.8 Usability

With personal networks, each person may have several embedded and wearable computers in addition to the more normal devices such as laptops and PCs. At the same time, we expect that personal networks should be for everyone and not only for experts and technology freaks. This is, in fact, the main target of our research; that anyone, including children, pensioners, and the sick can use personal networks. Furthermore, personal networks must support the user in his/her daily activities in an efficient and pleasant way. This means that usability is in fact one of the most important requirement for personal networks.

All personal network devices must be self-configured and able to communicate seamlessly with each other without requiring complicated user intervention. It must not be necessary for the user to configure any network settings, such as addresses or default routes. All networking solutions must be self-organized. Further, they must also automatically adapt to new situations without user involvement. Also the personal network applications should themselves detect their own settings and operate over any type of network in order to carry out their tasks. That is, a personal network and its applications must be able to operate without directions from the user. At the same time, this must be done in a secure way and with the user still in charge of all his/her devices. This is a deliberate problem and serious care must be taken when designing these types of systems.

The best way to achieve usability is actually to design a simple and obvious system, but still capable of fulfilling its other requirements. The idea is to be so easy, that the user easily can understand how it works and create an accurate mental model of the system. Whenever the system does not work according to the user's wish, he/she knows why and what needs to be done in order to make it operate correctly. In this way, the user will stay in charge of the system and the system will never do something unexpected. On top of this simple architecture, smart and complex components can be designed for certain specific issues.

Especially from a security perspective, a simple and understandable design is crucial [230][52]. Users who do not understand the system will probably not understand that they are vulnerable to malicious attackers and no user interface can perfectly hide such a flawed system design. In the worst case, the security system is so complex that the user becomes annoyed and tries to disable or circumvent the security system by, for instance, using simple or no passwords, allowing access to everything for everybody, or ignoring security warnings. It is therefore important that careful consideration is given to the issues related to usability and security; otherwise the risk is high that the user becomes the "weakest link".

2.1.9 Other requirements

These user requirements are by no means exhaustive. There are many more requirements depending on the target group, applications, and environment. In addition to user requirements, business-related requirements that various commercial personal network stakeholders may have are also not included as well as requirements related to legislations and regulations. We do not cover them, since those requirements are out of scope of this thesis. An attempt to cover a more complete set of requirements using user scenarios, use cases, and other approaches have been done by IST MAGNET [131].

2.2 Related Work

Many technologies have been proposed in the area of personal and wireless communication, but there have been very few attempts to achieve a complete and integrated solution for all personal communication issues. In this section, we list some earlier and current work aimed at either analyzing future personal communication requirements or building such integrated solutions. Only more complete attempts are listed here. A lot of work exists that addresses only one or a few aspects. It is clear that personal networks will build on many of those existing technologies. We will introduce those technologies in the later chapters where they are more relevant.

We do not consider IST MAGNET, IST MAGNET Beyond [130], Freeband PNP2008 [180], and IOP GenCom QoS for PN@Home [190] as related work. The research of this thesis has been a part of these research projects and hence their results are directly reflected in this thesis. The author of this thesis has been directly involved in the shaping of these projects. Hence, the work in this thesis is very similar to the work of those projects, but with slight modifications. Some alternative solutions to some aspects have been proposed in these projects. In later chapters, we will cover some of these variants and discuss pros and cons.

2.2.1 Ad hoc networking

Ad hoc networking is not truly a new topic. Its main foundation started decades ago under the term Packet Radio Network (PRNET) and was developed for military purposes [108]. Around 1996, the academic community started to show bigger interest in the topic with non-military applications becoming more important, such as emergency relief networks, home networking, community networks, and WPAN. From there, the research interest grew remarkably for some years, which made ad hoc networking a serious research area within wireless and mobile communication.

Ad hoc networking is all about quickly and automatically setting up an unplanned wireless network. Without configuration, wireless devices should be able to automatically find each other and establish a network for communication. When devices and their networks become mobile, this is even more important. Hence, the birth of the frequently used term Mobile Ad Hoc Networks (MANETs) [143].

A MANET consists of mobile devices with wireless communication capabilities that can move around freely and still communicate with each other in a distributed fashion and without external support. Due to limitations in

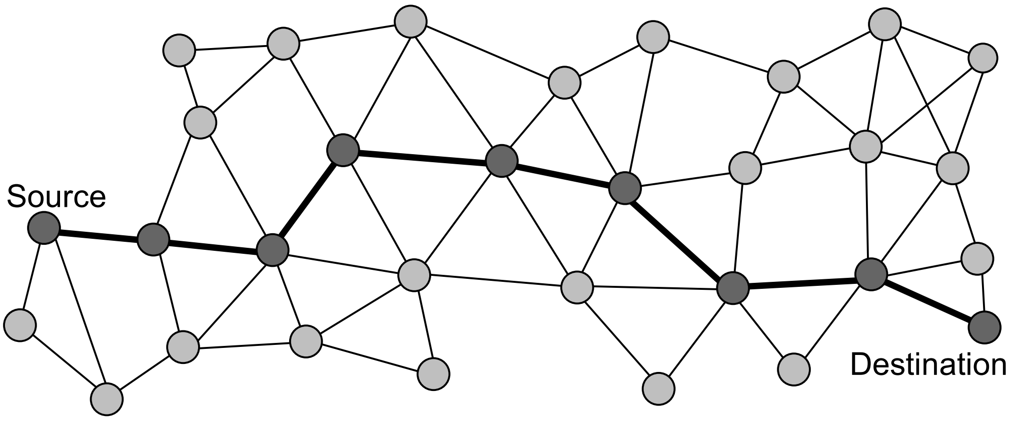


Figure 2.1: A Multi-hop MANET with a path

the utilized wireless communication technology, it may not be possible for two MANET devices to directly communicate. Instead, devices in between will assist by relaying the packets so that they can reach their final destinations. This is called multi-hop communication since a single packet must be retransmitted in several hops to reach its destination. Figure 2.1 shows a MANET that has a multi-hop path between a source and a destination.

When an ad hoc network is not fully connected, we need multi-hop communication. In order to find the hops that can connect the source and destination, routing is required. Several routing protocols have been designed specifically for multi-hop MANETs [177][143][26]. In fact, routing has been one of the most covered aspect of ad hoc networking and is still an active research area [32][40] because of the special characteristics of an ad hoc network in comparison with traditional communication networks. Examples of other frequently studied areas are network-wide broadcasting [231][215] and security [236].

From military and emergency relief networks to spontaneous conferencing and home networking, the far-reaching applications of ad hoc networking can transform the way in which network are deployed. It can offer networking in situations where a traditional network setup is impossible. Further, it is an ideal technology for personal networks as it offers self-configured, self-maintained, and self-organized networking. As a consequence, ad hoc networking will be an important building block for personal networks in achieving ubiquitous networking.

2.2.2 WWRF Book of Visions

The Wireless World Research Forum (WWRF) is a global joint-venture between academia, research institutes, and industry in the area of wireless communication. It includes manufacturers, network operators, service providers,

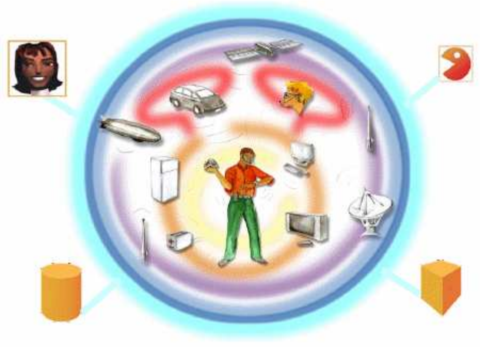


Figure 2.2: The MultiSphere model of WWRF Book of Visions (from [233])

and other related companies. The forum was founded in 2001 and provides a global platform for discussion of results, and exchange of views to initiate global cooperation towards systems beyond 3G. One outcome of this joint forum was The Book of Visions [233], that was first launched in 2000 [232] under the Wireless Strategic Initiative (WSI), which was a research project sponsored by the European Union. In the later years, The Book of Visions has appeared in book form under the name Technologies for the Wireless Future [217][218].

The idea behind The Book of Visions was to bring together the experts in this field to gather ideas and outline grand visions and challenges for the research and development of future wireless communication systems. Among a lot of ideas, the so called MultiSphere model was proposed to support further definitions and work on complex mobile communication concepts and ideas. Based on evaluation of some usage scenarios, this model was defined to reflect key characteristics of future communication scenarios. A graphical representation of the model is shown in Figure 2.2. The model is used to better understand the future of wireless communication and pinpoint areas that need more research.

The MultiSphere model acknowledges the importance of usability by placing the user in the center surrounded by various wireless communication systems that work together on behalf of the user. These systems were divided into spheres, where the PAN that connects a person's hand held and wearable devices is the innermost sphere. The next sphere consists of elements in the immediate environment with which the PAN devices can communicate. The third sphere consists of other near persons and more complex networks. The fourth sphere consists of the mobile networks (e.g., GSM and UMTS) as we know them today as well as future wireless wide area mobile networks. At this point, it is important to have interconnectivity among all the wireless technologies in the inner spheres and this is placed in the fifth sphere. Efficient and seamless integration is seen as very important since it must be possible for all wireless devices and persons to communicate with any other wireless device. The outermost sphere is the so called CyberWorld. Here, the wireless world meets the rest of the digital world, such as the Internet. In the CyberWorld, we may interact with smart agents, communities, and digital services.

Personal networks partly fits the MultiSphere model. Just as the MultiSphere model, it places the user in the center with wireless technologies around to support the user in his/her daily activities. Around the user, there is a PAN of personal devices which corresponds to the innermost sphere. These devices can interact with the immediate environment (Sphere 2 and 3). Further, a personal network uses infrastructure networks (Sphere 4) to extend the PAN to devices physically away from the user and his/her PAN (Sphere 5 and 6). Hence, personal networks will address many of the research issues identified by the WWRF in their Book of Visions, such as self-organization and integration of various network types, wireless service architectures, and more.

It must also be noted that the MultiSphere model does not cover all the aspects of neither personal networks nor any future wireless communication systems. E.g., it does not give much insight into human to human communication using wireless communication systems. Neither does it give any insights into security or privacy. Therefore, personal networks will not blindly adhere to the model. In any case, the WWRF model was never intended to cover all aspects.

2.2.3 Ubiquitous and pervasive computing and communication

While the WWRF Book of Visions has been mainly defined by the European telecommunications industry, ubiquitous and pervasive computing is a more American drive to future computing and communication and stems to a larger degree from the computer industry. The term was coined by Mark Weiser [229] of Xerox PARC in 1991. He noted that “people find a walk among trees relaxing and computers frustrating” and suggested to aim at making computing more ubiquitous, that is, intertwine computers in the everyday life and make them “vanish” into the background. Computing should be embedded into everyday objects to interact with and to enable people to move around. This would lead to enhanced user interactions since computers will disappear from our focus and become non-obtrusive. Devices should be tailored to suit particular tasks. Further, they should be enabled to sense changes in the environment and automatically adapt and act based on these changes as well as user needs and preferences. To be really usable, these ubiquitous computers also need to be interconnected so that they can communicate with each other and thereby cooperate to meet the needs of the users.

Ubiquitous computing is very different from personal networks in one major aspect. In the view of ubiquitous computing, computing devices are seen as commodity items that serve any user. They are meant to be shared by everybody. In fact, a personal laptop is even seen as a failure. No one should need to bring a computer as there should be ubiquitous computing objects available everywhere to be used by anyone. However, the current trend is towards an increased amount of personal devices. The most successful one is without doubt the mobile phone followed by laptops, personal digital assistants (PDAs), navigation systems, portable storage devices, and many more. Today, these devices are not only carried for their functionality, but also as personal attributes and status symbols. We love to have them and we love to personalize them by giving them their unique look and feel (and ringtones). The concept of personal networks acknowledges that some devices are personal and mainly used by one person. At the same time, there are ubiquitous computing devices in the surrounding that these personal devices can interact with.

It must, however, be said that much of the research that has taken place within the vision of ubiquitous computing and communication is very useful also for personal networks. This includes, but is not limited to, ad hoc networking, sensor systems and networks, and ubiquitous user-interaction.

Besides ubiquitous computing and communication, the terms pervasive computing and communication are also frequently used. Another, almost identical, vision is ambient intelligence. Ambient intelligence was first defined by the European Commission's Information Society Technologies Advisory Group (ISTAG) in 2001 [94] as a way to stimulate European research in this area. The Ambient Networks research project is an example of such research.

2.2.4 Ambient Networks

Ambient Networks (AN) [10] was a European research project that stemmed from the Wireless World Initiative (WWI), a spin-off from WWRF. It was an integrated project sponsored by the European Commission under the Information Society Technology (IST) priority under the 6th Framework Programme (similar to IST MAGNET and IST MAGNET Beyond). Its main objective was to create network solutions for mobile and wireless systems beyond 3G.

Most of the work carried out in this project concerns user devices and networks and their connections to access networks. Already today there are many different types of networks available to users and new network types are constantly being deployed. The main idea behind this project was to make each of these networks into ambient networks (ANs). AN offers a fundamentally new vision based on the dynamic composition of these ANs to avoid adding to the growing patchwork of extensions to existing networks. This will provide access to any network, through instant on-demand establishment of inter-network agreements. The Ambient Networks project was about taking this network cooperation a bit closer to reality.

A comprehensive prototype was developed within the project that gives users or networks the choice of using the appropriate radio technology automatically. They can switch between different flavors of 3G systems, WLAN, Bluetooth, or forthcoming 4G systems depending on what is the best network for a particular service or multimedia content. Systems for quality of service (QoS) sensitive multimedia services have also been developed.

Ambient Networks is more about the linkage between users' networks and infrastructure networks and between the different infrastructure networks than about the users' networks themselves. However, these links are still important for personal network communication and will support in reaching ubiquitous networking with QoS-support and reliability. Ambient Networks is therefore an important building block that can provide seamless infrastructure support to personal networks.

2.2.5 IST PACWOMAN and SHAMAN

Power Aware Communications for Wireless Optimised Personal Area Network (PACWOMAN) [171] and Security for Heterogeneous Access in Mobile Applications and Networks (SHAMAN) were two other IST projects that started slightly ahead of IST MAGNET. PACWOMAN worked mainly on WPANs and ad hoc networking. The networking environment was divided into three distinctive spaces [124][172]. The first space was the Personal Area Network (PAN), where personal devices can communicate with each other. The second space was the Community Area Network (CAN), which consists of nearby PANs belonging to different people that wish to interact with each other. The last space was the Wide Area Network (WAN), which provides each of the PANs with connectivity to remote devices. The main research area of PACWOMAN was link layer and medium access control for the PAN space. There, they identified the need of separating low and high data rate communication. Low data rate technologies are needed for sensors and other small devices with limited power. Such devices require simple and power-aware networking. Something that is highly useful for any type of personal networking.

IST SHAMAN [205] focused on providing a security architecture for PANs. The basis for their architecture was a trust model [64] that describes the basic security relations between different PAN devices (components in the SHAMAN terminology). Each device is owned by one user and that user determines, by means of security policies, who can use it. The security framework covers both local communication within a PAN and global access to the infrastructure. With respect to a particular device, other devices are classified either as first party devices, trusted second party devices, or non-trusted devices. A first party device has the same owner as the device itself and therefore has the highest trust. This model is implemented with a personal certificate authority (CA) [63] that runs on one of the owner's devices and hands out certificates in the form of public-private key pairs to all the user's PAN devices.

The work of both PACWOMAN and SHAMAN is highly relevant to personal networks and will undoubtedly provide an excellent foundation as they developed many important concepts for personal networks. PACWOMAN addresses part of the ubiquitous networking, device heterogeneity, and QoS requirements, while SHAMAN addresses usable security and trust requirements. However, we would need to combine the two into one solution.

2.2.6 Personal Distributed Environment

This project is part of United Kingdom's Mobile VCE [155] and has defined a concept called Personal Distributed Environment (PDE) [56][57]. PDE is a very similar vision to personal networks, but goes much further than just defining a vision. PDE is an attempt to define a concrete architecture and implement solutions that meets the vision. It is important to point out that the work of PDE and the work of this thesis have been going on in parallel. The proposed architecture and solutions sometimes overlap, but are very different in other aspects.

A PDE consists of a user's local and remote devices and services [204]. At the center of the PDE is the so called PDE server which has a Device Management Entity (DME) server component. The PDE server is basically a normal server somewhere in the Internet and the DME server component is a database of all devices belonging to a particular user. Each device in a PDE stays in contact with the PDE server to update its location, capabilities, and services in the DME. This connection can either be direct using a secure tunnel or by help from a proxy service on another device in the same PDE. In this way, it is possible for a device in the PDE to use services on any other device in the PDE through the PDE server.

While PDE focuses more on service delivery over heterogeneous networks, it also covers some network aspects. Devices within the same PDE that can communicate with each other using a local communication technology, form a so called "sub network". Examples of sub networks are PANs, Body Area Networks (BANs), home networks, office LANs, etc. Sub networks are then connected to each other through a core network using various access technologies and through the so called Network Access Devices (NAD).

The PDE assumes that each sub network already implements the necessary network and security solutions. These local mechanisms may differ between different sub networks and networking environments, but are kept unchanged. Instead, to make sure the PDE and its devices do not perform unauthorized tasks, a trust management system based on a trust engine that bridges the various trust and security systems in the various sub networks is proposed. This system is part of the DME and is also responsible for trust and security towards devices outside the PDE.

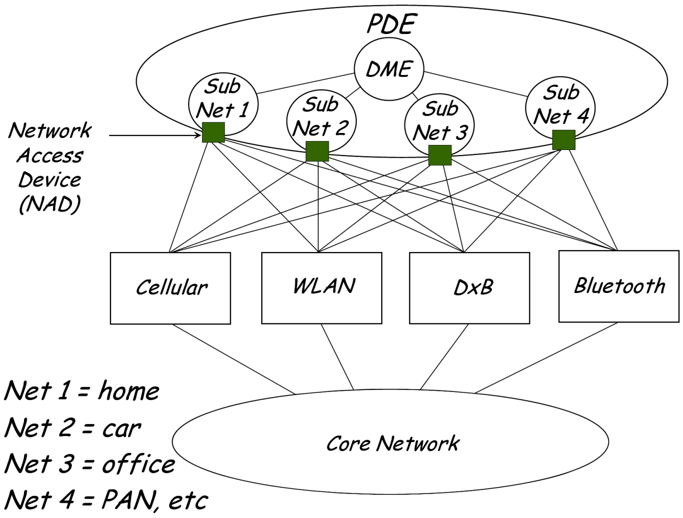


Figure 2.3: A Personal Distributed Environment with several sub networks (from [57])

PDE is, because of this, not a single homogeneous system, at least not when it comes to security and the sub networks. The benefit of doing this is to keep current solutions as they are and only provide integration when needed. However, it also makes the sub networks quite different from each other. The user may be burdened with several different technologies, configuration possibilities, user interfaces, and the like. Further, the sub networks will most likely not be compatible with each other and make it difficult for devices to move from one sub network to another. However, PDE addresses most of our requirements, except context awareness and privacy. From a usability and networking perspective, we believe it is better to provide one common security and network solution for all devices and networks in a personal network.

2.2.7 MyNet

The MyNet project [14] is a recently started project. It is a collaboration between Nokia and MIT and aims to study and develop a network architecture, tools and applications for simple, secure, personal overlay networks. The User Information Architecture (UIA) [107] and the Unmanaged Internet Protocol (UIP) [59] are projects within Massachusetts Institute of Technology (MIT), USA. MyNet is a collaboration project between Nokia research and MIT based on these projects. While there are prototypes for UIA and UIP, MyNet has just started.

UIP combines the self-management of ad hoc networks with the scalability of IP by creating a kind of self-organized overlay network for personal devices. UIA, on the other hand, is intended to allow global interaction and sharing among information devices between persons. The UIA protocols are

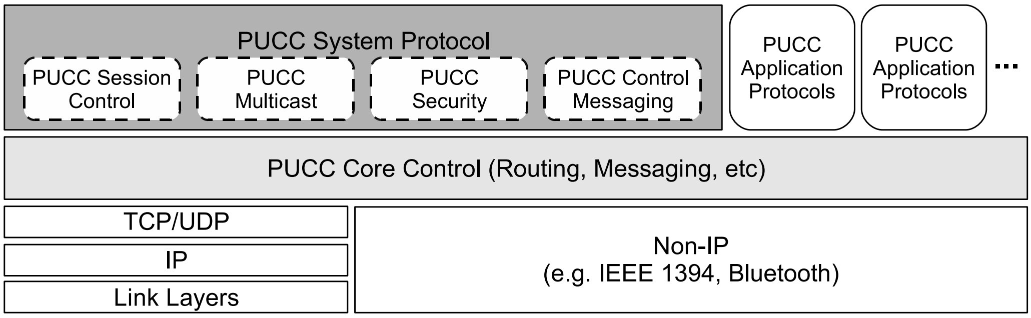

PUCC System Protocol
PUCC Session Control
PUCC Multicast
PUCC Security
PUCC Control Messaging
PUCC Application Protocols
PUCC Application Protocols
...
PUCC Core Control (Routing, Messaging, etc)
TCP/UDP
IP
Link Layers
Non-IP
(e.g. IEEE 1394, Bluetooth)

Figure 2.4: PUCC platform protocol stack

the foundation upon which the rest of the MyNet project work is layered. The UIA is based on two principles: (a) security is decoupled from physical connectivity; and (b) establishment of trust is based on social connectivity. This is achieved by creating personal and private name spaces. Simple to use mechanisms that leverage social relationships will allow a user to share access to their devices and resources using these name spaces.

These projects stem from the peer-to-peer research community, but are still highly relevant as they focus on many overlapping areas with personal networks. Security, ease of use, and self-organization are goals also for these projects. Hence, these projects do not fully support ubiquitous networking, but do address security and trust as well as naming management.

2.2.8 P2P Universal Computing Consortium

The P2P Universal Computing Consortium (PUCC) [188] is an university and inter-industry cooperation of some Japanese universities and companies active in Japan, such as NEC, Toshiba, and NTT DoCoMo. The target for PUCC is to realize a seamless peer-to-peer (P2P) communications technology platform that enables the creation of ubiquitous services between networked devices. The initiative has been going on since December 2004, but until recently, very little has been published.

The goal of PUCC is very similar to that of personal networks. With P2P overlays, they provide seamless communication between IP networks and non-IP networks such as home networks and sensor networks. A service platform provides seamless integration of services and other higher layer functionalities. However, the network layer is kept as is without any extra support. Figure 2.4 shows the proposed PUCC protocol stack.

Currently, protocol specifications have been proposed for most of the core functionalities as well as for key applications. Developer kits for networked devices are also being prepared. A demonstration of this was recently given at the Consumer Electronics Show of Las Vegas in January 2008. Despite the limited information published about PUCC, we believe it addresses many of our requirements, including service discovery and management as well as security. However, similar to MyNet, we do not feel the requirement of ubiquitous networking will be fully addressed by PUCC.

2.2.9 More related work

There are numerous other projects that touch on the aspects of personal networks. One early such attempt is Universal Personal Networking (UPN) [25]. UPN was a Siemens project in the early 1990s. At that time, WPANs were virtually non-existent and WLAN was very new. However, the aim of UPN was similar to the concept of personal networks, but the existing technologies at that time were a limitation. Hence, UPN focused on infrastructure-support for personal networking, device technology, and user interfaces. Taking place prior to the big break-through of the Internet, they spent a lot of effort on Internet-like techniques and security aspects were neglected. A more recent initiative from Siemens is their LifeWorks [118], which is a visionary concept of an unified communications experience for both business and private users. Under the umbrella of LifeWorks, Siemens develops products that aim at seamless convergence between fixed and mobile networks, new better services for mobile users, and ease of use. It is not only Siemens that works in this direction, but the whole telecommunication industry is showing an increased interest in this area. However, the current focus is more towards businesses and business services.

IBM defined and showcased a concept called Personal Mobile Hub (PMH) [79], which acts as a hub between a PAN and the infrastructure network. It can connect and control the PAN consisting of a person's wireless devices and also interconnect them to servers in the infrastructure. To demonstrate the concept, they developed a health-related application that monitored heart beats and blood pressure and alerted when certain thresholds were exceeded. Further, it could monitor that a person took his/her medication and otherwise alarm the person and/or caretakers.

In the academic world, it is also worth mentioning the work on personal networking by Robin Kravets' group at University of Illinois at Urbana-Champaign. Among the solutions they worked on, there is one called Mobile Grouped Device (MOPED) [114]. MOPED is a system that represents a person's set of personal devices as one entity towards the Internet using only one single Internet address. That address is given to a proxy node that is always available through the Internet. It is the task of the proxy to keep track of all the other personal devices and how they are connected to the Internet and to each other. Personal devices that can connect directly with each other form what they call components. The components may then connect to the Internet and the proxy. Hence, MOPED provides a technical solution to achieve personal networking and their focus is clearly on addressing, routing, load balancing, and mobility. While they solve many important aspects, there are many more still left open, such as security, support to higher layers, and direct wireless communication between MOPEDs.

There are many more smaller projects or specific solutions that target selected areas of personal networks. For instance, HP's CoolTown [43] gives people, places, and things a presence on the web. These dynamic web presences can then be related to each other to form new interesting applications that connects the virtual world with the real world. Standford's Mobile People [144] offers an application-level mobility solution for mobile persons, since it is the persons that are the end points and not the devices. They introduce a personal proxy that tracks the person and handles personal-level mobility aspects, including accepting incoming communication on the person's behalf, directing it to the correct device, converting the communication stream if necessary, and protecting the person's privacy. Both CoolTown and Mobile People are completely infrastructure-based and do not consider local communication and many other aspects of personal communication. However, they are useful as parts of the personal networks solution we are aiming for.

It is also worth mentioning that the Third Generation Partnership Project (3GPP) recently started to consider use-cases similar to personal networks in their drive towards All-IP networks (AIPN) [2]. In fact, they use the term "Personal Networks" for those use-cases, which involve a person with devices in different locations that are interconnected using 3GPP-networks as well as non-3GPP networks.

2.3 Related Work Requirement Analysis

In this section, we answer the question on how close the related work in the previous section come in terms of the user requirements. However, ad hoc networking, the work of WWRF, and ubiquitous computing and communication are more targeted towards steering research and development rather than producing something concrete. While research done within any of these areas certainly do contribute to personal networks, they do not propose concrete solutions. Hence, we leave them out of this related work analysis.

In Table 2.1, we list the more concrete approaches and how they address the various user requirements we listed in Section 2.1. A "Y" in the table indicates that a particular project do consider topics related to that particular requirement to a larger degree. However, it does not necessarily mean that the project fully meet that requirement. To the contrary, none of the projects meet all the requirements, which we already explained in the various subsections of Section 2.2.

Many of the requirements are covered by several projects, but no one covers all. The projects that have been running in parallel to this thesis work, PDE, MyNet, and PUCC, are the ones that come closest. Further, none of the projects cover context awareness. This is not to say that there

Table 2.1: Summary of related projects and requirements

	Ubiquitous Networking	Heterogeneous Hardware	QoS & Reliability	Naming & Service Mgmt.	Context Awareness	Security and Trust	Privacy	Usability
Ambient Networks	Y	Y	Y					
PACWOMAN	Y	Y	Y					
SHAMAN						Y	Y	Y
PDE	Y	Y	Y	Y		Y		Y
MyNet		Y		Y		Y		Y
PUCC				Y		Y	Y	
MOPED	Y	Y	Y					

are no context awareness projects. In fact, there are plenty, such as Freeband Awareness [15]. However, the only personal network-related projects that consider context awareness are MAGNET Beyond [130] and PNP2008 [180].

2.4 Summary

In this chapter, we formulated a small set of high-level user requirements that needs to be fulfilled in addition to the current state of art. The requirements include topics, such as ubiquitous networking, respecting heterogeneous hardware constraints, QoS, reliability, naming, service management, context awareness, security, trust, privacy, and usability. We argued that solutions that do not meet these requirements will not be able to meet the expectations from the users for personal networks. However, this set of requirements may still not be complete.

In the second half of this chapter, we listed technologies that have been proposed in the area of personal and wireless communication, including work aimed at either analyzing future personal communication requirements or building such integrated architectures or solutions. We also introduced major research areas, such as ad hoc networking and pervasive computing, from where personal networks has its roots.

We showed that very few attempts have been made to addresses all aspects of personal networks, leaving the users with only fragmented and incompatible solutions. Hence, we acknowledge that both more research and development are required; personal networks is a concept of great importance to unleash the full potential of wireless communications.

Chapter 3

The Personal Network Architecture

In the previous chapters, we introduced the concept of personal networks and what we expect from it. We also showed that there is no integrated solution to all of the user requirements, we expect a personal network to support. In this chapter, we propose an architecture for personal networks that supports or can support all the requirements outlined in the previous chapters. By architecture, we mean a high level description of the entire system, its components, their relations, interfaces, and main functionalities. The purpose is to bring structure, separate the various concerns, and make it easier to describe the system. The personal networks architecture is described at a high level with a focus on networking aspects. The main purpose is to give the reader a complete picture in order to better understand the remaining chapters and how the various solutions fit together.

It is important that the architecture does not violate any of the personal network user requirements or limit the possibility to fulfill all requirements. The complete personal networks architecture with its solutions must allow seamless communication among heterogeneous networks, work on mobile devices, offer reliable communication, be context-awareness, provide security and privacy, and allow the complete system to be easy to use for the users.

This chapter is organized as follows. Section 3.1 divides the architecture in three levels. Terminology for each of the levels is introduced and explained in Section 3.2, whereas Section 3.3 explains each of the levels. Then, attention is given to the network level since this is the focus of the thesis. Section 3.4 covers how new devices are introduced into a Personal Network. Networking aspects at a local level, i.e., how personal devices can communicate directly with each other without external assistance, is discussed in Section 3.5. Section 3.6 deals with ways to connect distant personal nodes over infrastructure networks, while Section 3.7 covers communication with nodes belonging to other persons. In Section 3.8, we discuss our proposed architecture's capability of meeting the requirements of personal networks. Section 3.9 concludes this chapter with a summary.

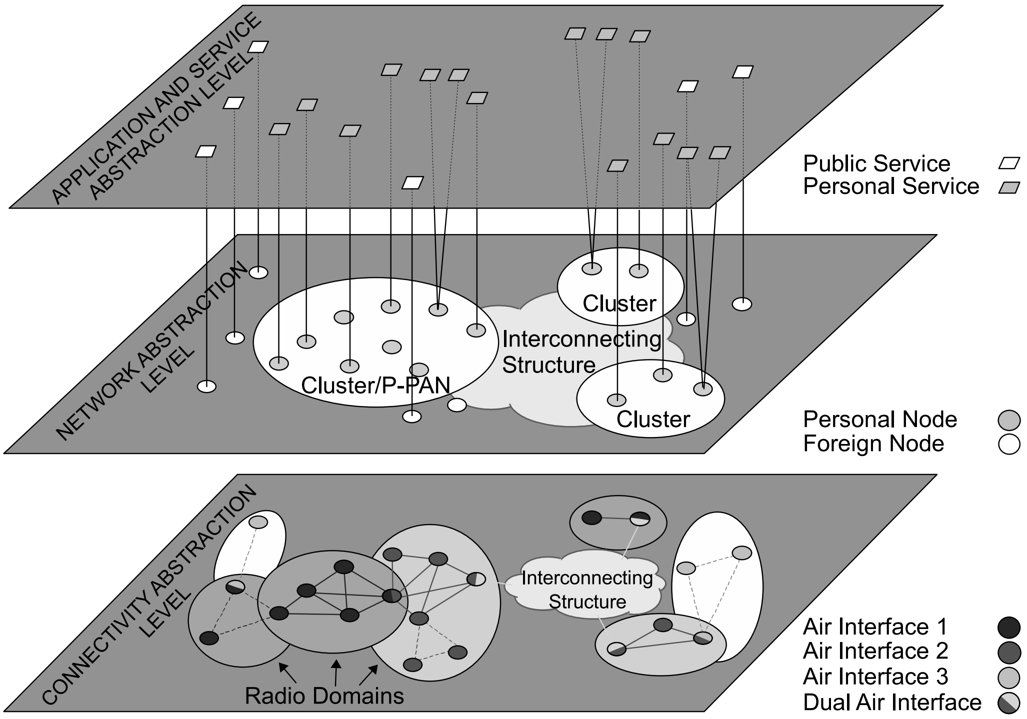


Figure 3.1: The Three Abstraction Level View of a Personal Network

3.1 The Three Level Architecture View

Our proposed architecture for personal networks is described using three levels of abstraction in order to better understand, structure, and localize the problems that need to be solved. It is a first step in developing the protocol solutions. Figure 3.1 shows the three abstraction levels and how they relate to each other.

Going from the bottom up, the first level is the *connectivity abstraction level*. Here, the devices can communicate with each other over common radio interfaces and Medium Access Control (MAC) mechanisms. The *network abstraction level* is placed above the connectivity abstraction level. In this level, routing and other networking mechanisms reside, which enable communication among all personal nodes. To reflect the provision and usage of services in the personal networks concept, the *application and service abstraction level* is defined on top. It contains all the applications and services offered by the Nodes in the network abstraction level. Only the applications and services are in practice visible to the user. Also less obtrusive services like name resolution and service discovery are part of this level.

The three abstraction levels view is a way to divide the architecture into three main parts for better understanding and decomposition of the problem to be solved. When it comes to implementation, a protocol stack has to be chosen and adapted to fit the abstraction levels. All popular protocol stacks are layered, such as the OSI reference model's seven layers [170] or TCP/IP's four layers [212]. In any case, the mapping of the layers onto the abstraction levels will be a rather straight forward task once we understand the details of each of the abstraction levels.

All these abstraction levels will be described in further detail in Section 3.3, but first a terminology will be developed that explains the concepts introduced at of each abstraction level.

3.2 Main Concepts and Terminology

In this section, we define the terminology used in the remaining of the thesis. The terms and concepts are similar to the ones defined in the IST MAGNET and MAGNET Beyond projects [133][153][98] and the PNP2008 project [181], which all are evolutions of [96] and [97]. However, some simplifications have been made since not all aspects of the architecture will be covered in this thesis. Most of the terminology related to the connectivity abstraction level and to non-IP devices have, for instance, been removed.

These concepts are the basis for the technical solutions for personal networks. As such, they are defined as precisely as possible. It is therefore important to point out that the definition of a Personal Network in this section is a technical concept that is of a very different nature than the way personal networks have been used earlier in this thesis. Previously, we used personal networks as a vision of what we are trying to achieve; from here on it will be a technical concept that tries to achieve that vision. When we use any of the terms defined in this section in the remaining of this thesis, we will write them with a capital letter to clearly indicate that we refer to the concepts defined here. Hence, *Personal Network* or *PN* refers to the term defined here, while *personal network* refers to the vision of personal networks as introduced in the previous two chapters.

Let us now introduce the concepts and their definition at the three levels of abstraction.

3.2.1 Connectivity abstraction level

Radio Interface A module that can send and receive data packets according to a particular radio technology and MAC mechanism. This term actually refers to all sorts of network interfaces, not only radio-based interfaces. The term was chosen because of the big interest in radio-based wireless communication and in order to be compatible with the terminology in MAGNET and PNP2008.

Radio Domain A collection of Radio Interfaces using a common communication technology that are controlled by a single MAC mechanism (either centralized, distributed, or a combination of both).

3.2.2 Network abstraction level

Node A communicating entity with one or more Radio Interfaces that implements the Internet Protocol (IP). It is understood that a Node also must implement the protocols and mechanisms required to form PNs as outlined in this thesis or in a future PN standard.

Trust Relationship Trust Relationships are established between Nodes that wish to communicate with each other. The Trust Relationship statues a certain degree of trust between the two Nodes. When Nodes come together and want to establish secure communication, they use the Trust Relationship they share, which is instantiated through a cryptographic mechanism. Trust Relationships can be permanent, which means they are valid until further notice, or ephemeral, which means they only exist for a limited time during which they are really used.

Personal Node A Node related to a given person (e.g., owned by a person) through a pre-established trust attribute defined by that person. The trust attribute determines to whom the Node is a Personal Node. Two Nodes with the same trust attribute are associated to the same person and are Personal Nodes to each other. All pairs of Personal Nodes have a permanent Trust Relationship between them.

Personal Network (PN) A Personal Network consists of Nodes sharing the same trust attribute with each other. That is, a PN is the collection of all a person's Personal Nodes and there are permanent Trust Relationships between all the Nodes in a PN.

Cluster A connected network of active Personal Nodes (usually located within a limited geographical area, such as a house or a car). Two Personal Nodes are in the same Cluster if they can communicate with each other using a path between them consisting of only Personal Nodes such that each Personal Node shares a single Radio Domain with the next Personal Node in the path. Hence, a Cluster consists of the Personal Nodes and the Radio Domains that connect them. A single Personal Node with no other Personal Nodes in its communication range is by itself a Cluster (a single Node Cluster).

Foreign Node A Node that has a different trust attribute and therefore is not part of the (same) PN. Foreign Nodes can either be trusted or non-trusted. Whenever trusted, they will typically have an ephemeral Trust Relationship with one or more Nodes in the PN.

Interconnecting Structures Public, private, or shared wired, wireless, or hybrid networks such as a UMTS network, the Internet, an intranet, or an ad hoc network that can be used to interconnect Clusters. We also assume that all Interconnecting Structures are connected to each other.

Gateway Node An active Personal Node within a Cluster that enables connectivity to Foreign Nodes and non-PN-enabled devices outside the Cluster. Some Gateway Nodes have access to Interconnecting Structures and can then connect to other distant Personal Clusters via the Gateway Nodes in those Clusters by means of inter-Cluster tunnels.

Personal Network Agent The Personal Network Agent or PN Agent is an infrastructure-based entity accessible through the Interconnecting Structures. Its task is to keep track of each Cluster and their attachment points to the Interconnecting Structure so that it can assist in establishing and maintaining inter-Cluster tunnels. It may also be an entry point for Foreign Nodes that want to use Services offered by the PN. Each PN has a PN Agent.

3.2.3 Application and service abstraction level

Service A logical component implemented by a software program running on a Node and that offers something that an application can use through a specified interface. Services have formalized descriptions that enable applications to discover useful Services. The description must describe what the Service offers, how to interface with it, and on which Node it is offered.

Client An application that can use Services. A Service discovery framework must be able to find the Services that Clients require. Clients can run on both Personal Nodes and Foreign Nodes.

Personal Service A Service offered by a Personal Node. The Service is therefore under the administrative control of the user.

Foreign Service A Service offered by a Foreign Node. The Service is not under the administrative control of the user, since it resides on a Foreign Node.

Private Service Private Services are Services accessible only within its PN. That is, the Node of a Client must have a permanent Trust Relationship with the Node of the Service. Furthermore, Private Services are never advertised outside its PN.

Public Service Public Services can be used by both Personal and Foreign Nodes. A Public Service may still require some sort of authentication or Trust Relationship between the Node of the Service and the Node of the Client. Public Services can be offered by both service providers as well as other person's Nodes. Public Services on Personal Nodes may be advertised outside the PN.

Service Management Node (SMN) A Service Management Node is a Personal Node in a Cluster that manages the Services and the Service discovery process for that Cluster. Every Cluster must have exactly one SMN. SMNs in different Clusters communicate with each other to provide Service discovery and management for the whole PN.

Service Proxy An application, running on a Gateway Node, which provides one or more Services available outside the PN to Clients inside the PN or the other way around by offering the exact Service itself. When calls come to the Service offered by the Service Proxy itself, it acts as a Client towards the real Service and just relays the Service interactions back and forth between the real Client and the real Service. Thereby, it can connect Clients and Services across the PN boundaries without requiring end-to-end network connections. It constitutes one way of doing foreign communication.

3.2.4 Other concepts

Private Personal Area Network (P-PAN) A Private Personal Area Network is the Cluster around the person. This term is primarily used to describe user scenarios and requirements. In earlier texts, such as [163], the term Core-PAN was used. Nevertheless, these terms have the same meaning.

3.3 The Three Abstraction Levels

3.3.1 Connectivity abstraction level

As mentioned earlier, we do not consider the connectivity abstraction level in this thesis. Nevertheless, an accurate abstraction of this level is still needed in order to correctly design the higher abstraction levels and to understand the connections and interactions with the lower levels. This level is composed of devices with Radio Interfaces connected to each other via different Radio Domains corresponding to given radio technologies. Figure 3.2 shows an example of devices with Radio Interfaces participating in several separate Radio Domains.

Problems addressed at this level concern link layers, medium access control (MAC), the physical link, and their interrelationships. Solutions for this abstraction level can rely on existing technologies, such as WLAN [82] and Bluetooth [84], but also future technologies, such as IEEE 802.15.3 [85] and the new air-interfaces developed in IST MAGNET and MAGNET Beyond

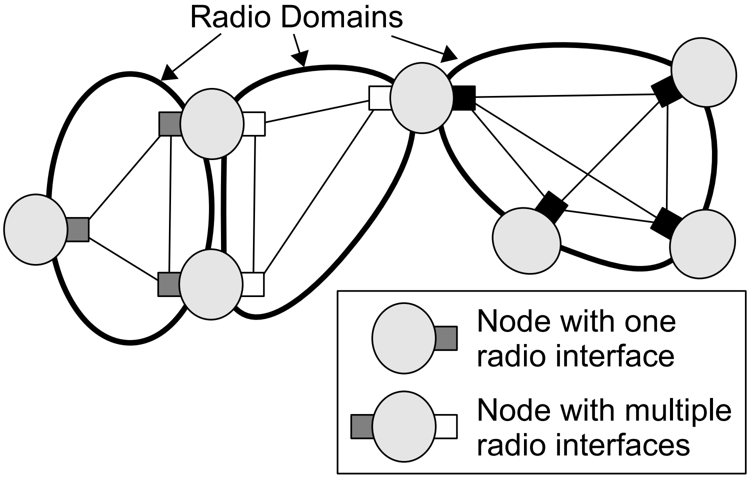

Radio Domains
Node with one radio interface
Node with multiple radio interfaces

Figure 3.2: Nodes with Radio Interfaces communicating through Radio Domains

[201][202]. Since, it is virtually impossible to adopt one single radio technology for PNs, we believe that many different kinds of radio technologies will be utilized. Some are suitable for short range high speed communication, while others are more suitable for access to the Interconnecting Structures. Furthermore, radio technologies will continue to evolve. Since older Nodes may use older technologies, it is important to enable gradual shifting from older technologies to newer ones.

At the same time, these different radio technologies may be so different that they will not be able to interoperate with each other at the connectivity abstraction level in any meaningful way (e.g., IEEE 802.15.1 [84] and 802.15.3 [85]). Therefore, we need devices with multiple radios and a network abstraction level that can glue all these technologically different radio technologies, both current and future, together.

3.3.2 Network abstraction level

Concepts, such as Clusters and PNs are defined at the network abstraction level. Given a single user, there are two types of Nodes at the network level: Personal Nodes and Foreign Nodes. Personal Nodes are Nodes that belong to the user, while all other Nodes are Foreign Nodes. The PN is the collection of all that user's Personal Nodes, both remote Nodes and the Nodes, which are in the close vicinity of the user. Active Personal Nodes are grouped into Clusters depending on their possibility to communicate with each other without external assistance. A user is likely to have several active Clusters, such as for instance: a home Cluster, an office Cluster, a car Cluster, etc. The Cluster directly around the user is also called the P-PAN and can consist of carried and wearable Personal Nodes. In other words, the PN is an extension

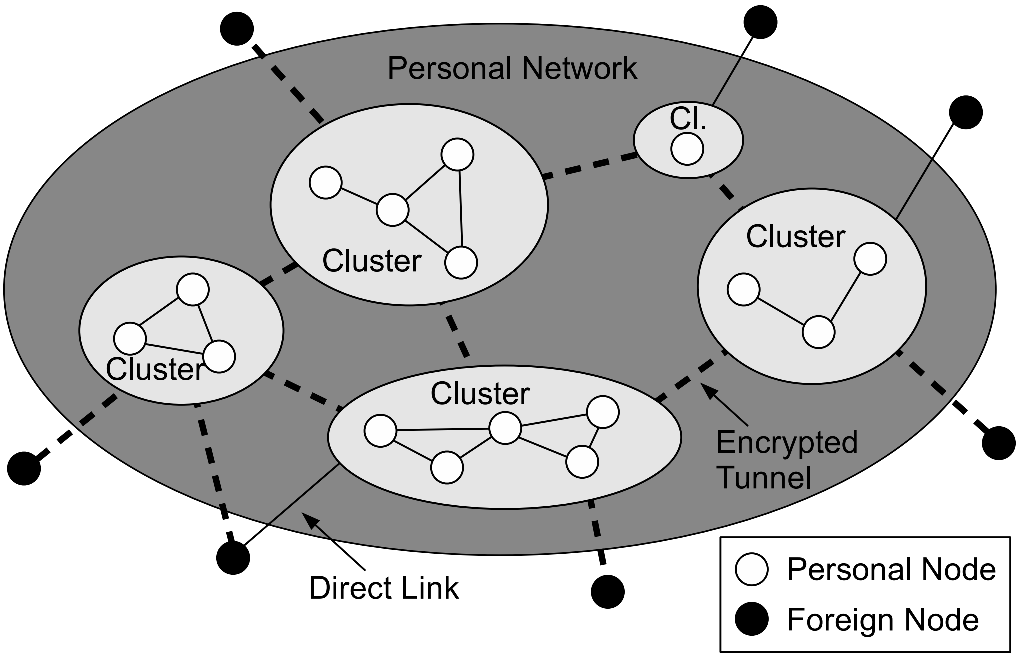


Figure 3.3: PN Network Level View

of the P-PAN and may contain several active Clusters, both remote and in the vicinity of the user. See, for instance, Figure 3.3.

The network level architecture separates the communication among Personal Nodes of the same PN from the communication to, from, and among other Nodes and devices. To make this happen, each Node must know which PN it belongs to and must be able to tell if another neighboring Node is a Personal Node or not (i.e., belongs to the same PN). The process of introducing a Node into a PN is called *personalization* and is a prerequisite to any Cluster and PN formation mechanisms. Personalization should only take place once when a new Node is acquired for the first time by the user. A Node then "permanently" remains a Personal Node until the user decides otherwise. Which Nodes the user chooses to personalize is obviously up to the user, but it could include Nodes he/she owns or otherwise possesses for an extended period of time.

All personalized Nodes become Personal Nodes to that user and each Personal Node establishes mutual Trust Relationships with all the other Personal Nodes. The personalization mechanism is responsible of establishing all the necessary Trust Relationships among all the Personal Nodes. Automatic mechanisms are needed to efficiently implement the personalization of new Nodes. The Trust Relationships must be maintained in an accurate, secure, easy to use, and efficient way so that two Personal Nodes always will be able to communicate securely. Before a Node changes owner, is discarded, or when a personalized Node is compromised, the personalization should be invalidated by the user. All Trust Relationships with a compromised or excluded Node must be removed immediately. Section 3.4 further discusses the personalization procedures.

The next step is to enable the Nodes to distinguish Personal Nodes from Foreign Nodes among its neighbors. This is the first step of the Cluster formation and should be fast and light-weight as well as secure against currently known attacks (such as replay and man-in-the-middle attacks) and denial-of-service (DoS) attacks. When some Personal Nodes find each other, they can start to exchange routing information and other kinds of information and thereby form a Cluster consisting of only Personal Nodes. Foreign Nodes are kept out of these mechanisms to protect the organization of the Cluster. Section 3.5 discusses Cluster organization further.

Not all the Personal Nodes will be able to communicate with each other in this way due to the characteristics and limitations of the radio technologies available to Cluster Nodes. Instead, a PN will consist of several Clusters at various remote locations. The only way to connect the different Clusters is to use Interconnecting Structures (such as the Internet). To this end, tunnels between the Clusters will be established and maintained. In Section 3.6, we cover more about these inter-Cluster tunnels and other PN organization aspects.

Though a lot of communication will take place within the PN among a user's Personal Nodes, it is also important that communication can take place between PNs and with non-PN devices. This we call *foreign communication*; this topic is covered in Section 3.7.

3.3.3 Application and service abstraction level

On top of the network abstraction level, we find the application and service abstraction level. This level consists of applications running on the Nodes that offer Services to each other as well as using these Services. Whether the Node where a Service is running is a Personal Node or not also influences this abstraction level since it sits on top of the network level. We make a distinction between Personal Services and Foreign Services depending on whether the Service is offered by a Personal Node or a Foreign Node.

Further, a Service can be Public or Private. Private Services are offered and used only by Personal Nodes in the PN sense. This implies that these Services can only be used by the person him/herself within the PN. It is important to realize that the protection offered by the fact that a Node is a Personal Node is not always satisfactory as the actual user of that Personal Node still is unknown to the Service. Some very sensitive Private Services may therefore require extra security such as authentication of the real end-user, for instance, by using passwords or biometrics.

On the other hand, Public Services are offered by the service owner to anyone that the owner wants to share the Services with. Public Service can

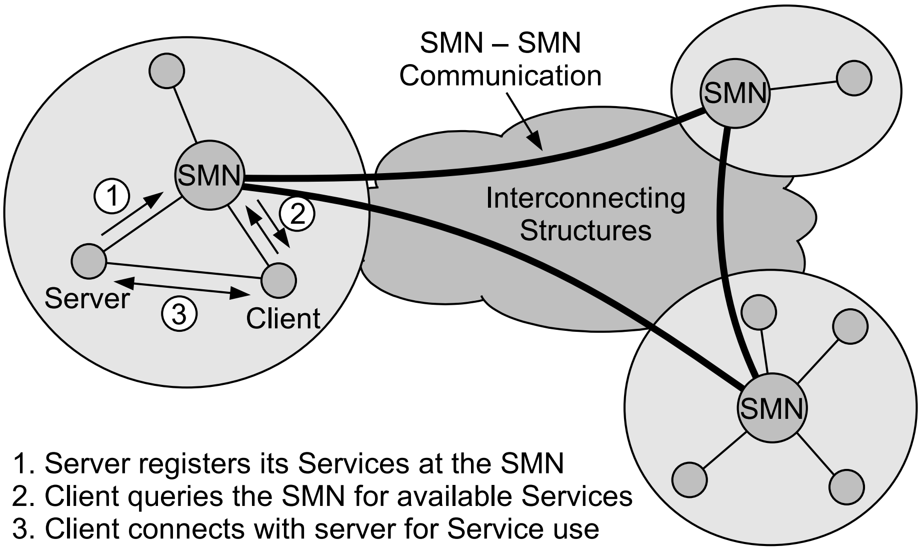

SMN – SMN
Communication
SMN
SMN
SMN
SMN
Interconnecting
Structures
1
2
3
Server
Client
1. Server registers its Services at the SMN
2. Client queries the SMN for available Services
3. Client connects with server for Service use

Figure 3.4: Service Discovery in a PN

be offered by someone's PN as well as by commercial providers. Many Public Services do not require any Trust Relationship, while others may still require establishment of an ephemeral Trust Relationship between the Service Node and the Client Node. In some cases, service usage may be charged.

To further improve the usability of a PN, a service discovery and management framework is defined that can support the applications and enable auto-configuration and adaptation. Figure 3.4 shows how this PN service framework works. Each Cluster has a Service Management Node (SMN), which is elected among the Cluster Nodes and works as a repository where all Services are registered. Clients in the Cluster that wish to use a particular Service can query this repository. To also facilitate usage of Services across Clusters, the SMNs in the different Clusters communicate with each other and share Service information. The SMNs can also advertise the Public Services of the PN to surrounding Foreign Nodes as well as register Foreign Public Services in case a Client on a Personal Node wishes to use a Foreign Service. In addition to service discovery, the SMNs may also control the ongoing service sessions to better manage the service provisioning. However, service discovery and management are beyond the scope of this thesis. For further information, see [134][65][66][139].

Another important functionality at this level is naming and name resolution. The use of names is crucial in order to build a user-friendly PN; a requirement identified in Section 2.1.4. Basically anything that needs to be seen by the user should have its own name, such as Nodes, PNs, Services, and perhaps even Clusters. The naming scheme must provide a flexible naming resolution to the PN and at the same time be compatible with the naming schemes used in the current infrastructure, e.g., the Domain Name System

(DNS) [148]. However, naming solutions for PNs are beyond the scope of this thesis, but have been proposed by others [154]. For inter-PN naming solutions, the MyNet project may provide an excellent solutions [107].

3.3.4 Interaction between the levels

Mechanisms in the various levels need to interact with each other in many different ways. The more information that can flow between the different levels, the better each of them can operate. This is usually known as cross-layer information and includes information from the physical layer and all the way up to the application layer. However, not only cross-layer information within the Node itself is useful, but also context information coming from neighboring Nodes and the environment can sometimes substantially improve certain decisions.

To better manage this information, a cross-layer and context information management framework should be deployed. The details of such a framework are out of scope of this thesis, but ongoing work is currently trying to design and evaluate such frameworks for PNs. See, for instance [113][20][139].

3.4 Personalization of Nodes

A core concept of a PN is the distinction between Personal and Foreign Nodes. The obvious next question is how to make a Node into a Personal Node, i.e., how is the personalization of Nodes done?

There are many different ways this could be done. It could for instance be based on Subscriber Identity Module (SIM) cards, similar to the ones used in mobile phones today [3]. Each user has a couple of SIM-cards that share some common secrets. By inserting a SIM-card into a Node, it becomes Personal. However, this would require the user to get these SIM-cards from somewhere and all PN Nodes to be equipped with SIM readers. Instead, a more realistic personalization process is based on the pairing of Personal Nodes. Let us describe this pairing process.

The first Node a user buys must be manually configured by the user. The PN is given a name and perhaps a unique PN identifier to distinguish it from other PNs. Other settings can be configured at this time, such as the address of the PN Agent (which will be described in Section 3.6). The next step is to pair other Nodes that the user possesses with this initial Node or other already personalized Personal Nodes.

In the pairing process, the Node to be personalized establishes a secure connection using a shared Radio Domain to an already personalized Node. Using this connection, they exchange pair-wise keys, i.e., they establish the required permanent Trust Relationship. These keys are later on used to securely detect each other and to establish secure connections also in the future.

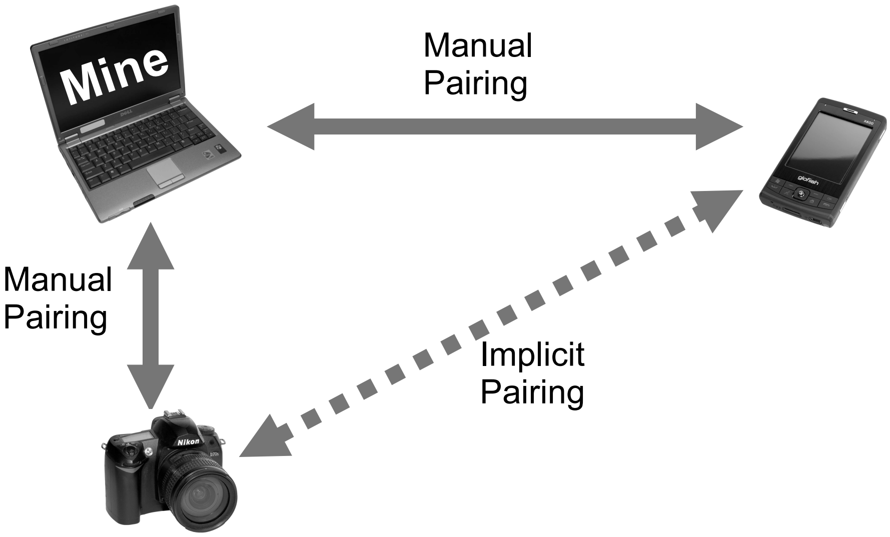


Figure 3.5: Pairing of Personal Nodes

At the same time, important configuration settings can be given to the newly personalized Node. The next step is to ensure that Trust Relationships (i.e., pair-wise keys) are also established with all the other Personal Nodes in the PN. To avoid the need for the user to manually pair the new Node with all other Personal Nodes, an implicit pairing procedure is defined that automatically pairs the new Node with all the other Personal Nodes. This procedure takes place without requiring the user to do anything extra. Figure 3.5 shows these steps.

The end result of the personalization of a Node is that it is fully included in the PN. It becomes a Personal Node. As explained before, it establishes a Trust Relationship with all the other already personalized Nodes in the PN. This means that the new Node gets access to the other Nodes in the PN as well as gives access to the other Nodes in the PN. To be precise, it gets and gives access to the PN networking mechanisms. Hence, the new Node can generate and trust routing messages, hello messages, data packets, etc. Since only Personal Nodes can participate in this, only Nodes that are under the user's control are included in these mechanisms. Therefore, all Nodes in the PN can be assumed to be cooperative. In this way, many security-related problems with respect to networking are solved as we protect the entire network from unauthorized devices.

It is of utmost importance that the pairing process is secure. At the same time, it must not require too much hardware and software support from the Nodes and must be easy to use. Proximity authenticated channels (PAC), which requires the two Nodes to physically be close to each other, cables, or removable media (e.g., flash memory cards, USB pocket memories, and compact discs) can provide increased security during the personalization and the exchange of the pair-wise keys compared to using the normal Radio

Interfaces. Several schemes have been worked out to improve the security of this procedure [138].

It is also important to prevent unauthorized pairing. Just physical access to a Personal Node for a short while must never be enough for someone to perform unauthorized pairing. In that case, it would be too easy for someone to include an unauthorized Node into the PN without the consent of the PN owner. Hence, the personalization must be protected by passwords, biometrics, or other secure means. Furthermore, if a Node is already personalized and belongs to a PN, it must not be trivial to remove or alter that personalization. However, this thesis is not going to cover this topic any further, instead we assume the Trust Relationships are in place. The cryptographic details of the pairing process and other aspects of personalization have further been described in [138].

3.5 Cluster Organization

At the connectivity abstraction level, Personal Nodes that have direct radio connectivity (a common Radio Domain) and that share a permanent Trust Relationship can establish secure communication. When more Personal Nodes are discovered, with which secure communication can be established, a Cluster is formed. A Cluster consists of only Personal Nodes and the secured communication links between them. Once secure connectivity has been realized, communication at the network level can take place between all Personal Nodes in the Cluster, without using Foreign Nodes.

Each step of the mechanisms to form and maintain a Cluster should be fully distributed. The proposed method is pro-active and opportunistic in its way to form Clusters. It uses all opportunities to find all neighboring Personal Nodes and thereby making the Cluster as large as possible. This means that the Cluster can handle its internal communication by itself without any outside support. We refer to this as intra-Cluster communication. To solely rely on intra-Cluster mechanisms as much as possible is likely to be more efficient than using inter-Cluster mechanisms. Inter-Cluster mechanisms involve an Interconnecting Structure, which is not always available and is, in many cases, expected to be poorer (in terms of performance, cost, etc.) than intra-Cluster communication. This is due to the longer distances that a infrastructure-based network needs to cover in order to be available everywhere. See Figure 1.1 in Chapter 1.

Clusters are dynamic in nature. Nodes are switched off or become available as well as roam and show up in different Clusters. Clusters can split when a person brings some Nodes and leaves the rest behind. Likewise, Clusters can merge when a person arrives home with his/her wearable Personal Nodes and they merge with the home Cluster. Potentially, there is no limit to how large a Cluster can grow, both in terms of number of Nodes and geographical span. However, typically we expect Clusters to have a small number of Nodes and a limited geographical span, because of the way they will be deployed. Typical radio technologies used for intra-Cluster communication will have limited range (e.g., IEEE 802.15.3 has a typical range of 10 m), which means that normal Clusters will be limited.

Not all Nodes in a Cluster will have direct links to all other Nodes, because of different link layer technologies or radio range limitations. This implies that a Cluster might be a multi-hop network. Therefore, it must provide addressing and routing functionality in order to enable efficient and secure communication. This functionality should be able to deal with the specific characteristics and dynamics of Clusters: Nodes can roam, join, and leave the Cluster. In order to handle these characteristics and dynamics, a distributed, totally self-organized, and efficient multi-hop mobile ad hoc routing mechanism should be used.

The Cluster around the user is the P-PAN. The P-PAN operates with similar mechanisms as the Clusters. Due to the dynamic behavior of the Clusters, Personal Nodes in the P-PAN or any of the Clusters may roam and join other Clusters or the P-PAN. It is therefore natural that all Clusters of a PN, including the P-PAN, use the same Cluster organization mechanisms for better integration when nodes roam or Clusters merge and split. There is, therefore, no need to handle the P-PAN differently from the other Clusters at this level. Because of this, we will use the Cluster to denote any Cluster or the P-PAN in this thesis. Only at the application and service abstraction level it may make sense to distinguish the P-PAN from the other Clusters, because the P-PAN is the Cluster closest to the user.

The use of the Internet Protocol (IP) as a common language makes it possible to have a network layer architecture that is independent from the heterogeneity of the underlying link layers. In this thesis, we will mostly focus on IPv6 [48], which is slowly being adopted [240]. The prototyping in the later chapters are based on IPv6, except the inter-Cluster tunnels that use IPv4, since they have to go across the Internet as it is deployed today. Nevertheless, most of the mechanisms of this thesis will also work over IPv4 [184].

Clusters can also consist of very limited devices that have no IP capabilities whatsoever. Examples of such devices can be networked sensors and simple actuators. As these devices still can offer important services to other Nodes in the Cluster and the rest of the PN, they should still be connected. Therefore, a Personal Node that can act as a bridge between the IP-incapable device(s) and the Cluster should enable this. The bridge can make the services offered by the IP-incapable device(s) accessible to the rest of the Cluster and the PN.

Finally, a Cluster will not only operate as a stand-alone network, but it will also interact with its immediate environment, such as nearby Foreign Nodes or the Interconnecting Structures. Nodes in the Cluster that can provide connectivity to Nodes outside the Cluster are called Gateway Nodes. Gateway Nodes will have some special functions such as address translation, filtering of incoming traffic, set up and maintenance of inter-Cluster tunnels, etc. These tasks might be quite heavy for some Personal Nodes, so it is useful to select powerful Personal Nodes as Gateway Nodes when possible. The process of finding capable Gateway Nodes with links to Foreign Nodes or the Interconnecting Structures is another network function that is provided by the Clusters.

Chapter 4, 5 and 6 will further discuss these and other issues related to Cluster organization.

3.6 PN Organization

A PN can have multiple Clusters that are geographically dispersed. As stated earlier, each of these Clusters uses the same mechanisms, but they organize themselves in a completely standalone fashion. The PN concept realizes communication between these remote Clusters using the Trust Relationships that are already deployed. With remote, we mean that communication between the Clusters can only be realized through the use of IP routing and forwarding over Foreign Nodes, such as through an Interconnecting Structure. Another implication is that once the Clusters have access to an Interconnecting Structure, they need to be able to locate each other, a fundamental property that forms the basis of PN organization.

In order to form the PN, three requirements need to be fulfilled by the PN organization mechanism:

1. When access to the Interconnecting Structure is available, the Clusters need to be capable of locating each other.

2. Once they have located each other, they must establish secure tunnels between them.

3. Last but not least, once the PN has been formed, it should be able to maintain itself in view of the dynamic nature of the networks. That is, the tunnels must be updated when one or more Clusters roam.

As mentioned before, Gateway Nodes are Personal Nodes in the Clusters that have connectivity with Foreign Nodes or the Interconnecting Structure. It is the responsibility of these Nodes to construct the interconnecting communication needed to build a connected PN and thus have to look for opportunities to establish such communication. In order for each Cluster to locate the other Clusters in the PN, an agent is of a big advantage. This agent is referred to as a PN Agent and each PN should have one. Its role is to coordinate the Clusters and keep their locations in a database. In this way, Clusters within one PN can easily find each other. Figure 3.6 shows a

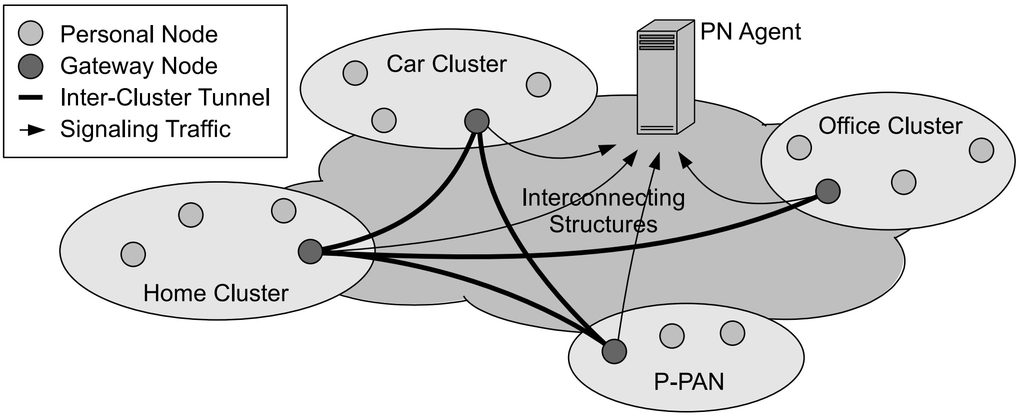


Figure 3.6: PN Organization

PN with a PN Agent assisting the Clusters to connect with each other using inter-Cluster tunnels between Gateway Nodes in the different Clusters. The PN Agent should be considered as a functional concept and not as a PN entity, as there may exist many different solutions to implement the PN Agent concept, including distributed solutions.

The purpose of the tunnels is twofold:

1. They provide secure means for inter-Cluster communication by shielding the intra-PN communication from the outside world.

2. These tunnels will be established and maintained dynamically to efficiently deal with Cluster mobility. Optionally, they can also help connecting Clusters that resides behind network address translators (NATs) or firewalls.

Once the PN has been formed, intra-PN communication can take place. However, in order to establish connectivity among the Personal Nodes, addressing and routing are indispensable. One possible approach is to see the PN as a single large multi-hop ad hoc network in which most of the links are wireless, some are wired, and some are tunnels between the Clusters. Within this ad hoc network, we can adopt a flat addressing scheme and run an ad hoc routing protocol that has been optimized for this environment. For instance, a PN internal IP prefix could be reserved and all Nodes within the PN will select a PN-unique IP address with this prefix. This IP address will be independent of the location of the Node in the PN. This approach has the great benefit that, in combination with the dynamic tunneling mechanisms, mobility will become completely transparent for the higher layer protocols. The ad hoc routing protocols will hide intra-Cluster mobility and the dynamic tunneling will hide Cluster mobility and Gateway Node changes.

In Chapter 7, we discuss the PN organization further, including various PN Agent solutions as well as some other alternative solutions for PN organization.

3.7 Foreign Communication

So far, only communication between Personal Nodes has been covered. However, a PN cannot exist in isolation, but needs to interact with other PNs as well as PN-unaware Foreign Nodes and other non-IP devices. Foreign communication involves both using Services from Foreign Nodes as well as offering Services to these Nodes.

At the connectivity abstraction level, at least one Personal Node must obviously share a common Radio Domain with the Foreign Node to be able to establish any communication. If this is the case, then the network level must provide the PN with a way to also communicate with Foreign Nodes without compromising its own security and adversely influence the intra-PN mechanisms.

As stated earlier, Personal Nodes that connect to a Foreign Node are called Gateway Nodes. Gateway Nodes need to treat foreign traffic in a different way from intra-PN traffic. They must, for instance, block all non-approved traffic from entering the PN. Furthermore, Gateway Nodes must bridge the mechanisms used inside the PN with the ones used to communicate with the Foreign Nodes as these mechanisms will be different. If a Personal Node instead wishes to communicate with a remote Foreign Node through an Interconnecting Structure, then the Gateway Node that links the Cluster up to the Interconnecting Structure needs to bridge the PN-internal mechanisms with the mechanisms used on that Interconnecting Structure.

When a Foreign Node wishes to establish communication with the PN also when no direct local connection is available, it can turn to the PN Agent of that PN via the Interconnecting Structure. Foreign Nodes need only to remember the address of the PN Agent of a PN to be able to initiate connections with that PN. To simplify even further, the address of the PN Agent can be given a name that can be resolved through, for instance, DNS. The PN Agent will know the location of all the Clusters in its PN and can tunnel the packets to the appropriate Cluster and Personal Node. At the same time, the PN Agent will bridge between the Interconnecting Structure mechanisms and the intra-PN mechanisms.

Chapter 8 contains more details about foreign communication.

3.8 PN Architecture Discussion

Now that we have introduced our architecture for PNs, we should ask ourselves two questions:

1. Can this architecture be implemented in reality, and

2. Does it then fulfill the requirements outlined in Chapter 2?

The first question will be answered in the following chapters where each part of the architecture will be further discussed and developed to a level where the architecture becomes concrete. However, answering the second question is actually more important, since if the architecture does not meet the requirements, then all the development is for nothing.

It is hard to know in advance whether the requirements can be met by the architecture since it depends on the detailed solutions chosen for each part of the architecture. For instance, QoS and reliability depend heavily on the used routing protocols and mobility mechanisms, which are not dictated by the architecture. Therefore, we can not say whether the requirements have been met. However, it is important that the architecture does not prohibit us from meeting the requirements. In the remainder of this chapter, we will try to answer the question by discussing some known issues and their potential solutions.

3.8.1 Why a network layer overlay?

The first question one must ask about the architecture is: Why define the PN at the network level and in effect creating a network overlay? An option would indeed be to define the PN at the service level, by perhaps building a service overlay on top of existing network solutions. While this certainly is possible, it would leave many important issues unsolved at the network level. The users would still need to configure many things; assign network addresses, routing protocols and other network settings for all kinds of different networks. Current auto-configuration solutions are either targeted towards only one network type (such as a home network) or are non-sufficient (e.g., cannot handle mobility). It is also likely that different types of devices use different types of incompatible network solutions that would make integration unnecessarily hard.

In the PN architecture we defined, a unified network solution is proposed that enables automatic configuration and adaptation. It provides connectivity between Personal Nodes and is protected from non-cooperative Nodes. Also foreign communication is possible.

3.8.2 How protected is a PN?

The main concept of this architecture is the strong focus around the long-term trust concept, which is used to make the distinction between Personal and Foreign Nodes. Only Nodes that have established long-term Trust Relationships (i.e., Personal Nodes) can be part of the user's PN. In this way, the intra-PN network mechanisms within a PN can be protected from non-trusted Foreign Nodes.

As we explained before, when a Node is being personalized, it establishes a long term Trust Relationship with all the other already personalized Nodes in the PN. This actually means:

1. The new Node gets access credentials from the other Nodes in the PN.

2. The new Node gives access credentials to the other Nodes in the PN.

The next question to be answered is of course: what exactly does this access include?

One essential capability it gets and gives access to is the intra-PN networking mechanisms. The Node can generate routing messages, hello messages, data packets, and so forth that are trusted by its neighboring Personal Nodes as well as verify the authenticity of such messages it receives. Since only Personal Nodes can participate in this, only Nodes under control of the user are included in these mechanisms. Hence, the Nodes can be assumed to be cooperative and in this way, many security-related problems with respect to networking can be avoided. Personal Nodes can send packets among themselves and be sure that packets they receive originated from a Personal Node. Packets from Foreign Nodes are either filtered or are treated in a way so that an end Node can distinguish them from packets coming from Personal Nodes.

Another capability a Personal Node gets and gives access to is Services. That is, a Personal Node allows access to its Services from any other Personal Node. The benefit of this is obvious. This access control policy is very trivial which hopefully means that most users can easily understand it. On the other hand, it has serious security drawbacks. If this policy is always used, then anyone can use a Personal Node to access any Service in the whole PN and this can happen since long term Trust Relationships only authenticate Nodes but not the user of a Node.

The problem includes lost or stolen Personal Nodes as well as Nodes temporarily in the hands of someone else. If such a Node is capable enough, it can be used by anyone to access any Service within its PN, including sensitive Services such as private photo albums, banking services, etc. The owner can of course exclude lost or stolen Nodes from the PN, but to only depend on this is not necessarily enough since it may already be too late. However, for many Personal Services, this weak level of security may still be acceptable, but since Personal Nodes certainly will be lost or stolen, there is also a need for more security for more sensitive Services.

A potential solution is to make sure that each Personal Node authenticates its direct physical user before the Node can be used to access Services on other Personal Nodes. However, then the security data used to implement the long-term Trust Relationships and stored on each Personal Node needs to be properly protected. This requires good user authentication and good tamper-resistance on all Personal Nodes. Unfortunately, good tamper-resistance for every Personal Node is a very difficult problem and currently not feasible

[211]. It is expensive and makes devices bigger, clumsier, and may still not yield good enough security.

As a consequence, to allow a Personal Node access without the Service itself authenticating the user is not secure enough for sensitive Services. For these Services, additional authentication and access control on the Service Node itself is required. Hence, it must be possible for the user to add security measures when he/she feels the need for it.

3.8.3 How usable is the PN Security?

One of the big concerns with the architecture is its security system and whether the users can handle it properly. Many other security systems, such as Pretty Good Privacy (PGP) [230], have failed just because users do not understand them. As before, it is impossible to answer this question without first implementing the system and conduct a usability study. Nevertheless, we can make sure that the architecture follows design principles known to work well, such as the ones outlined in [237].

The design principles regarding usable security identified by Yee can be summarized as follows: explicit authorization, visibility, revocability, path of least resistance, expected ability, appropriate boundaries, expressiveness, clarity, identifiability, and trusted path. Some of these principles, such as path of least resistance, identifiability, and trusted path, have more to do with the user interface than the system architecture itself.

We believe that the other principles either are supported or can be supported by the final implementation. Personal Nodes are manually included into the PN by the user (explicit authorization) and can again be excluded (revocability). Since each Node needs a Trust Relationship with all other Personal Nodes, it would not be difficult to create a tool that displays this information and tells the user exactly which Nodes are currently part of the PN (Visibility). The remaining principles (expected ability, appropriate boundaries, expressiveness, and clarity) have to do with what the security system is capable of and how this is communicated to the user.

The distinction between Personal Nodes and Foreign Nodes can be based on, for instance, ownership, which is a concept people understand. The system is, at the same time, visible. Hence, the user can easily understand how it works and therefore create an accurate mental model of the system more easily [166]. With the help of the mental model, abilities and boundaries become clear and the user will be able to take the right decisions.

3.8.4 Do we need to manage our PNs?

Today, private persons have problems to update and secure their own home PC. To install, update, and maintain sophisticated devices that are heavily software based, e.g., digital home theater systems, hard disk recorders, digital cable or satellite tuners, etc. cause considerable dissatisfaction. What will happen when they have tens or maybe hundreds of wireless devices to maintain? The idea is that PNs are self-configured and hence minimizes the need for manual management. For many of the management task required by today, it is clear that further technical solutions are required. As many of these tasks as possible must be automated by the PN to minimize the demand on the user to manually perform these tasks.

Automatic software updates and frequent backups of Personal Nodes are examples of tasks that a PN management tool must provide. This should not be a problem with the current architecture. These types of tools need to communicate with all Personal Nodes and perhaps a few Public Services through the Interconnecting Structures, something that is already supported. With these connection possibilities, automatic solutions can be built as well as remote management solutions if necessary. Further, the system should inform the user when manual actions are required, such as switching on a deactivated Node when there is a need for a software update. However, there is a need for a streamlined and common framework for this, considering that a PN may consist of several very different types of Nodes from different manufacturers with very different types of software.

3.8.5 What about the social dimension?

The real challenge with personal networking is actually to understand the social aspects and here it must be agreed that the PN architecture is not fully sufficient. For instance its focus on the person and his/her Personal Nodes makes it difficult to share Nodes. Most people have affiliations with a lot of different communities where Nodes actually are shared. A person might belong to a family, a company, a sports organization, as well as informal groups of friends and acquaintances. The architecture must take these facts into account and properly support individuals in their social life. For instance, borrowing and sharing of Personal Nodes must be supported, maybe only for a limited time in some cases, but also indefinitely. An example of the latter is a family sharing home equipment and appliances.

Occasional and exclusive borrowing of a Node can be done by removing all sensitive information on the Node and give the borrower full control of the device. That is, all stored data should be moved to another Personal Node and the Trust Relationships temporarily be deleted during the time someone borrows it. The Node can be excluded from the owner's PN and temporarily included in the borrower's PN. Renting is special form of occasional time-limited borrowing where this scheme also could be used. As long as there is only one concurrent user and the user does not change too often, this will work.

However, another solution is required when it comes to sharing a single Node among several persons at the same time. Relying on foreign communication is of course possible, but not always efficient. A shared Node that is frequently used by two persons will never be part of both PNs and this may lead to sub-optimal networking for at least one of the persons. An alternative is to allow a Node to be personalized by two or more persons simultaneously, but still keep the various PNs separate. A Node with several personalizations need to be able to distinguish among the PNs and then maintain several concurrent but separate intra-PN networking mechanisms. This may be a bit heavy for mobile Nodes, but is not a problem for stationary and mains-powered Nodes that are more likely to be shared like this anyway.

Sharing networks can also be beneficial. Consider, for instance, a home network belonging to a family. Such a network consists of several Nodes and network links that should be shared by all the family members. It is not efficient if each of the family members have their own totally separate home Cluster. It would be more efficient if their Clusters integrate into one family home Cluster. After all, Nodes belonging to family members can also be believed to be cooperative. It may therefore be beneficial if some PNs can be fully integrated at the networking level. Two PN users can, for instance, declare each others PNs as totally trusted. Whenever a Personal Node meets a Foreign Node belonging to a fully trusted PN, it considers that Node to be a Personal Node, at least at the network level. At the service abstraction level, the Node can still be considered as a Foreign Node to make sure that Services only available to Personal Nodes are not accidentally opened up.

Group communication is another area where the current architecture is somewhat limited. Considering that group communication is both common and important for people, it can be worthwhile exploring alternatives. One particular promising solution is federation of networks (Fednets) [164] and in particular federation of PNs [75]. The idea is to extend the PN framework with special functionality for group communication. This includes establishing and maintaining federations of PNs in an efficient and ad hoc way to support collaborative tasks of PN users. A group of users that needs to communicate establishes a PN federation among their PNs and makes relevant Services available within that federation. The details of how a PN federation could work have, for instance, been specified within IST MAGNET Beyond [130] and Freeband PNP2008 [81].

One more important social aspect is the conflict of interests between a person's professional and private life. On the one hand, companies set up security regulations on how an employee can use the company devices and how to handle sensitive company data and on the other hand, that person has a private life where he wants to use the same devices. This may include having a combined agenda for both professional and private activities. It may be desirable to be able to take care of some private matters when at office or using privately owned equipment when working from home, etc. However, this will undoubtedly go against most company's security regulations if PNs can not prove to be secure enough. Companies need to trust both the PN architecture and their employees' ability to use the system in a secure way before they allow their employees to have one PN containing both private and business devices at the same time.

3.8.6 More issues?

As has been demonstrated here, the base architecture outlined earlier in this chapter is far from perfect. It tries to single out the trustable Nodes from non-trustable Nodes, but sometimes fails to do so. Personal Nodes can be lost and can therefore not always be fully trusted, while at the same time, many Foreign Nodes actually are trustable and can be fully trusted. The architecture is very simple and easy to understand, but not yet sufficient. With this, we hope to have highlighted that these issues can indeed be solved. The current strength of this architecture is its simplicity. However, extensions are not only possible, but also required even if they make the PN more complicated. It is important to carefully consider which extensions are useful so that we can avoid creating an unnecessary complicated system. However, in the rest of this thesis, we will focus on the feasibility of building a PN based on the base architecture and leave this for future research.

3.9 Summary

In this chapter, we proposed an architecture for personal networks in which there are two types of nodes: Personal Nodes and Foreign Nodes. Personal Nodes are nodes that belong to the user, while all other nodes are Foreign Nodes. The Personal Network (PN) of a user is the collection of all that user's Personal Nodes. When active Personal Nodes come together and can communicate with each other, they form Clusters. Personal Clusters communicate with each other over Interconnecting Structures and thereby form a connected PN. The architecture separates the communication among Personal Nodes of the same PN from the communication to, from, and among other Nodes and devices. Hence, each Node must know which PN it belongs to and must be able to tell if another neighboring Node is a Personal Node or not. This is achieved by a manual personalization step in which the user admits a newly acquired Node into the PN.

To better introduce the various concepts of the architecture, we divided it into three abstraction levels: the connectivity abstraction level, network abstraction level, and the application and service abstraction level. Terms and concepts were then introduced at each of these levels.

In the last part of this chapter, we discussed the suitability of this architecture. We focused on the requirements introduced in Chapter 2 and verified whether the architecture could fulfill all requirements. Certain areas, such as security, usability, and social aspects, were studied in more detail. It was concluded that most requirements were fulfilled or could be fulfilled by our architecture. However, due to its simplicity, the architecture is not fully sufficient; some extensions are required. However, the rest of this thesis will focus on the feasibility of building a PN based on the base architecture and leave this for future work.

Chapter 4

Cluster Formation and Maintenance

Now that the overall architecture for PNs has been determined, it is time to look at the various parts of the architecture. In this chapter, we start by looking at the Clusters in more detail and their formation and maintenance in particular.

In Chapter 2, we argued that personal networks need ubiquitous network connectivity that is reliable. To achieve this, it is best if the Personal Nodes can manage their own communication among themselves without the need for infrastructure support, at least at the network level. However, if there is no need for special communication equipment, such as access points and network switches; this will maximize the ability of ubiquitous network connectivity.

Each Personal Node should contain the necessary functionality to form and maintain the networks needed for its operation. Hence, when Personal Nodes come together, they should form a local ad hoc network that enables them to communicate with each other independently of external support. They should form a self-organized Cluster.

This chapter is structured as follows. Section 4.1 precisely defines what a Cluster exactly is and introduces important requirements for the Cluster formation. Section 4.2 discusses related work, with a focus on the technologies we propose to use to implement Clusters. Section 4.3 introduces the Cluster formation mechanism and its neighbor discovery mechanism. The security-related topics of Cluster formation and maintenance are covered in Section 4.4 and 4.5. Section 4.4 covers Personal Node authentication while Section 4.5 covers the establishment of secure communication with neighboring Personal Nodes. In Section 4.6, we outline the Cluster formation prototype that we developed to demonstrate the concepts of Cluster formation and maintenance. Section 4.7 summarizes this chapter.

4.1 Definition, Scope, and Requirements

A Cluster is a connected network consisting exclusively of Personal Nodes located within a limited geographical area, such as a house or a car. The Nodes share Trust Relationships with each other and are connected by one or more link layer technologies [135]. All Personal Nodes should implement the same mechanisms at the network level so that they can easily find each other and communicate directly when there is connectivity between them at the connectivity level. In this way, the migration of a Personal Nodes from one Personal Clusters to another becomes easier.

A Cluster facilitates secure network communication among Personal Nodes on a local scale and sets a boundary for the intra-Cluster network level formation and maintenance mechanisms. A precise definition of a Cluster is needed in order to clearly set a boundary between local and global network mechanisms. We therefore define a Cluster as follows:

Definition: A *Cluster* is a connected network consisting of active Personal Nodes exclusively.

Two Personal Nodes are in the same Cluster if they can communicate with each other using a path between them consisting of only Personal Nodes such that each Personal Node shares a Radio Domain with the next Personal Node in the path. Hence, a Cluster consists of Personal Nodes and the Radio Domains that connects them. A single Personal Node with no other Personal Nodes in its communication range is by itself a Cluster, a single Node Cluster.

It is important to note that neighboring Nodes within a Cluster share Trust Relationships. Each Node must validate whether a neighbor is a Personal Node or not. This is done by cryptographic means based on the preestablished Trust Relationships between the Personal Nodes. Personal Nodes within a Cluster are usually located within a limited geographical area, such as a house or a car. This is due to the limitation of typical link layer technologies that we envisage will be used in a Cluster, such as LAN, WPAN, and WLAN technologies. Long range infrastructure technologies, such as UMTS, can not be used within a Cluster. Such technologies rely on several Nodes not under the control of the Cluster owner and can never be characterized as a single Radio Domain with one common MAC mechanism. Instead, they can be used when a person's Clusters need to communicate with each other.

There are of course alternative ways to define a Cluster, such as using geographical distance. A Cluster can be defined as Personal Nodes within a distance of 10 meters. This definition requires positioning capabilities and may not be suitable for network layer mechanisms even if approximating techniques are used (e.g., based on distance measuring using received radio signal strength). However, geographical proximity among Nodes within a

Cluster can be a good indication for security. Two nodes in the same Cluster will most likely reside in the same situation, such as inside a house, a car, or on the body of the PN owner. Indeed, security based on this kind of information may be useful, but should instead use mechanisms at higher layers, such as a context information distribution functionality. Another possible Cluster definition is a limit on the number of hops from an elected master node or a special device (e.g., [216]).

Cluster formation and maintenance at the network level will take place in the same way in every Cluster, including the P-PAN, and should therefore comply with the same set of requirements. We see the following requirements as crucial:

1. The network layer and the Cluster formation and maintenance procedures must be as independent as possible from the lower layers, including MAC and link layers. At least current link layer technologies must be supported, including Ethernet (IEEE 802.3 and 802.11), Bluetooth (version 1.2 and 2.0), and future link layer technologies as far as can be anticipated. While wireless is more convenient for the user, we should not completely forget wired solutions either.

2. A Cluster can be connected using various link layer technologies. The Cluster formation and maintenance functions must support multiple different link layer technologies in a single Cluster at the same time.

3. The Cluster formation and maintenance must be self-organized. It must be able to form and maintain itself without support from the user or external infrastructure-based equipment or services.

4. Personal Node discovery and departure detection should be provided by the link layer. However, when no such functionality is available or not adequate, the network level has to provide this functionality. For improved quality of Cluster networking, it is important that links between Personal Nodes are constantly monitored so that link breaks and Node departures can be detected in an accurate and timely manner.

Scalability is not a very important requirement for Cluster formation and maintenance. We believe that, in the future, mobile Clusters typically will not consist of very large numbers of Personal Nodes. Non-mobile Cluster, such as a home Cluster, may contain more Nodes; perhaps up to around 50 Nodes. There are plenty of electronic devices in a house that can benefit from being connected in a permanent way. We expect that very large numbers of devices, such as sensors, will need special solutions and therefore be “hidden” behind single Personal Nodes acting as gateways between the PN and the sensor networks. Therefore, we design the Cluster formation and maintenance procedures to support up to at least 100 Nodes, but with a typical size of less than 10. Scalability is mainly an issue in intra-Cluster routing.

4.2 Related Work

A Cluster is basically a special type of ad hoc network [177]. Ad hoc networks and their mobile variant, MANETs, were described earlier in Section 2.2.1. As mentioned there, MANETs are unplanned networks of mobile devices that can form quickly and automatically when the opportunity arises. As such, Clusters will build on ad hoc and MANET technology.

In addition to routing, the IETF MANET working group [143] has recently started to focus on other subjects. New topics include a common packet format for signaling traffic in MANETs [38] and a neighbor discovery protocol called MANET Neighborhood Discovery Protocol (NHDP) [39]. The new common packet format is extensible and has been extended for carrying the signaling of future ad hoc routing protocols, such as OLSRv2 [40] and DYMO [32]. In the same way, it is possible to extend the packet format to also carry Cluster-related information, such as Trust Relationship information. However, one aspect not covered by the packet format is privacy. While it is possible to encrypt some signaling traffic, such as routing traffic, it is not possible for every signaling packet. Routing traffic is exchanged between Personal Nodes only after discovery and when a secure link has already been established. Hence, these packets can be completely encrypted by the link layer and there is no problem with using the common packet format in combination with any ad hoc routing protocol. On the other hand, Node discovery packets and Personal Node authentication packets can not be encrypted in the same way, since they are transmitted before a link has been secured. Therefore, they require special mechanisms to both securely authenticate the Nodes and at the same time protect the privacy and the identity of the Nodes. Therefore, we will use a dedicated packet format for these packets in this thesis.

The neighborhood discovery protocol (NHDP) from IETF MANET, which of course is based on the common packet format, uses periodic hello messages. Each Node periodically transmits hello messages on all its MANET-enabled interfaces. The hello messages contain the Node's address, a list of its neighbors, and some other basic information. The purpose of the hello messages is two-fold: first, they are used to discover new neighbors. Whenever a hello message is received on an interface from a previously unknown neighbor, a new link can be established. The second purpose is to monitor existing links and detect when they disappear. However, as will be explained in Chapter 6, this raises several issues not addressed by NHDP. For instance, using hello messages provides a very inaccurate view of a link's quality and is a slow way to detect changes. In Chapter 6, we will provide a better view of the link qualities. Further, we will propose solutions to the privacy problems that NHDP will have.

The Ananas approach [33] tackled the MANET problem by introducing an adaptation layer between the link layer and the network layer to emulate a normal switched Ethernet for the network and higher layers. This "layer 2.5" actually uses ad hoc networking techniques to implement unicast and broadcast packet forwarding on top of a multi-hop network. The benefit of such an adaptation layer is that higher layers can continue to operate as if they were running on a normal fixed network. The drawback, on the other hand, is that too much information is hidden, which leads to inefficient and non-optimal operation. Protocols designed for switched Ethernet environments do not work particularly well in ad hoc networks as they are not able to cope with the special requirements that an ad hoc network poses, such as varying link quality and lower available bandwidth. However, the main problem is actually the dependency on broadcasting. Broadcasting in a switched Ethernet is both a reliable and relatively inexpensive operation. In an ad hoc network, that is not the case. Therefore, Ananas' model is not robust and efficient enough for Clusters.

The idea to introduce a new layer for ad hoc networking between the link and network layers has been proposed by many others for other reasons as well. One popular reason is to provide a single abstraction of a link. Generic Link Layer (GLL) [197][198] and Universal Convergence Layer (UCL) [137][199] are both variants of that approach. The idea is to avoid that the network layer needs to be developed to work with a particular type of link layer or to accommodate all the details of all the different types of link layer technologies. The reasons are on the one hand, a network layer needs to be independent from the link layer to allow it the flexibility to use any type of link layer, and, on the other hand, ad hoc networking requires a good amount of cross-layer interaction [241][242][243]. A generic or universal abstraction of an ad hoc link layer by an adaptation layer can offer a good compromise between the two. The Cluster formation mechanism will therefore use the same kind of approach in order to be able to work with several different types of link layer technologies.

4.3 Cluster Formation

The method we propose for Cluster network formation is opportunistic in the way it forms a Cluster. It seizes all opportunities and makes the Cluster as large as possible using available ad hoc link layer technologies. These technologies do not have the same range as mobile networks that may cover a whole country or more. On the other hand, they have much higher bandwidth for the same energy consumption. This makes them preferable for use in Clusters when compared to infrastructure-based mobile networks. The Cluster formation can be thought of as a network sub-layer function that makes use of available link layer technologies to form a Cluster whenever possible. It is also proactive, meaning that it constantly looks for Personal Nodes in the neighborhood and possible links to extend the Cluster. The

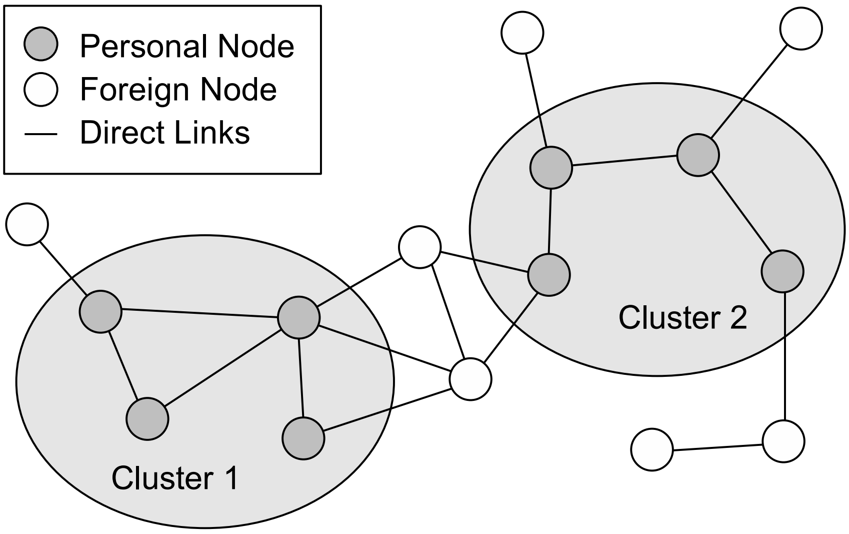


Figure 4.1: Example of two Personal Clusters

purpose is to be able to use intra-Cluster mechanism to provide communication between as many Personal Nodes as possible, since local communication is likely to be more efficient than using global Interconnecting Structures. Interconnecting Structures cannot be assumed to always be available and, in many cases, provide lower data rates and at higher cost than direct multi-hop intra-Cluster communication.

After a Cluster has been formed, each Personal Node will have a list of direct neighbor Nodes that are Personal Nodes. Associated with each neighboring Personal Node in this list is also a security key, which is used to protect the data messages on the link between the two Personal Nodes. The security key is of course derived from the pair-wise keys that were deployed during the personalization step. Traffic from neighboring Foreign Nodes must be ignored unless the Node is acting as a Gateway Node. Gateway Nodes treat traffic from Foreign Nodes differently with firewall rules and the like. Properly encrypted traffic from neighboring Personal Nodes, on the other hand, is always trusted and, in this way, a secure Cluster is formed. A Cluster carries traffic among its Personal Nodes in a secure way and detects and filters traffic from Foreign Nodes. On top of these secured links and filtered neighbor lists, we introduce an intra-Cluster networking layer with addressing and routing. Note that also these functions will be protected in the same way as any traffic between neighboring Personal Nodes. Figure 4.1 shows an example where several Personal and Foreign Nodes are scattered in a small area. The Personal Nodes will find each other, establish secure links, and form Clusters. Due to radio range limitations, it is impossible to form one Cluster containing all Personal Nodes. Hence, in this scenario, two Clusters are formed and to connect the two, we depend on an Interconnecting Structure.

4.3.1 Multi-hop Clusters

Each of the steps to form and maintain a Cluster is distributed. This is because we do not allow the Cluster formation to depend on special Nodes being present. Any two Personal Nodes that implement a common wireless technology must be able to find each other and form a secure Cluster. Furthermore, distributed solutions are more robust, especially in multi-hop networks and Clusters may be multi-hop. That is, a Cluster may not always be fully connected with direct links between all of its Nodes. Even though Clusters will be fully connected in many cases, we cannot exclude the possibility that a Cluster is multi-hop and hence the Cluster formation and maintenance procedures need to support multi-hop Clusters. In particular, there are three reasons that may make a Cluster multi-hop:

1. Some link layer technologies have limitations on the number of devices in one PAN. The best example is Bluetooth [84], which can only accommodate eight active devices. If more active devices are present, a multi-hop scatternet [147] is needed in order to accommodate additional devices.

2. The radio range may not be enough to cover all devices in one hop. Imagine a house where a person's devices are spread all over that house, including home appliances. In principle, a wired WLAN infrastructure with several access points could be deployed in such a situation, but there is actually no need for that. It is also possible to connect the devices in a multi-hop ad hoc network. If necessary, one can place mains-powered relay devices just to make sure that the multi-hop Cluster is connected and covers the whole house.

3. It is necessary to accommodate several link layer technologies at the same time, e.g., the different variants of IEEE 802.15 together with 802.11. If two nodes want to communicate but do not share a common radio technology, then a third device that implements both can act as a bridge; this is in fact a kind of multi-hop network. Obviously, it would be easier for the user to have only one single link layer technology, but in times of technology shift this will make a gradual change of technology possible. Another reason for multiple radio technologies in one PN is their diverse characteristics. Some technologies offer high data rate while others offer low power consumption and unfortunately, no single link layer technology satisfies all application and device requirements.

This means that we need to cater for multi-hop Clusters and hence need multi-hop routing. However, the routing protocol should be optimized for small Clusters that are fully or almost fully connected.

4.3.2 Link layer device discovery

The first step towards forming a Cluster is to detect neighboring Nodes and their hardware and IP addresses. There are three different ways this could be achieved:

1. Use the device discovery mechanism provided by the lower link layers. Several link layer technologies implement device discovery mechanisms. IEEE 802.11 uses beacons carrying the MAC-address to detect neighboring nodes and access points. Bluetooth transmits inquiries to detect neighbors and form piconets. These messages can also be used by the Cluster formation to detect new neighbors or the disappearance of neighbors. The advantage is that no new signaling is needed. However, different link layer technologies use different device discovery mechanisms and it is therefore necessary to define how new devices are detected for each possible link layer technology. Furthermore, it is possible that some link layer technologies do not have any device discovery mechanism that can be used by the Cluster formation mechanism. Wired Ethernet is an example of that. In that case, other device discovery mechanisms are needed.

2. Use the neighbor discovery [158] of IPv6 [48]. Unsolicited neighbor advertisement messages may be used to advertise a Node's presence. This mechanism is mainly used to map IPv6 addresses to hardware addresses, but may also be used for detecting new neighbors. However, every Node needs to transmit a neighbor advertisement periodically and thus adds signaling traffic. IPv4 does not have such a mechanism.

3. Implement a higher layer device detection mechanism. Hello messages based on the user datagram protocol (UDP) [183] sent to an IPv6 link local multicast address would be a good option. By defining a new multicast address for IPv6 Nodes that understands PN mechanisms, it is possible to filter out non-PN-capable Nodes. These messages could also make the mapping between the IPv6 address and the hardware address just like the IPv6 neighbor discovery messages above. A similar approach is possible for IPv4 as well.

The best solution would be to use the link layer device discovery when possible and otherwise use a higher layer mechanism. The link layer discovery option does not usually offer the mapping between hardware addresses and the IP addresses. However, this is not necessarily a problem since the Trust Relationship detection will take place immediately afterwards and the mapping can be part of that procedure instead. Figure 4.2 shows how a Personal Node that has different link layer technologies with different amount of support could work. The leftmost Radio Interface uses a link layer technology that has a discovery mechanism that can be used almost completely, only a

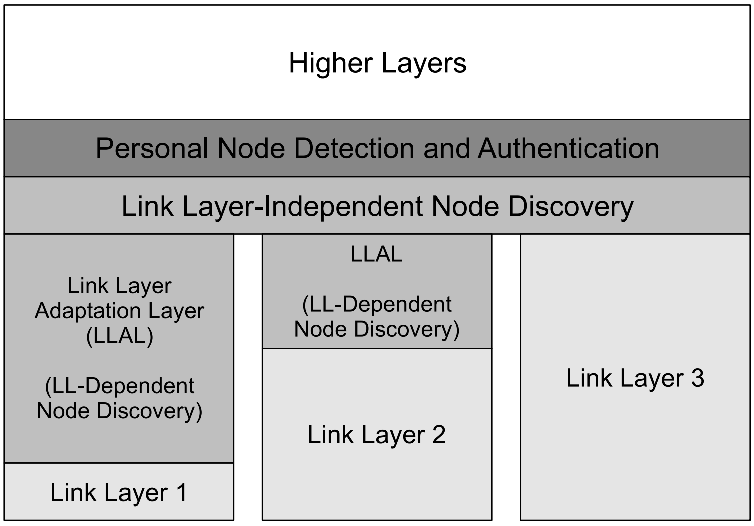

Higher Layers
Personal Node Detection and Authentication
Link Layer-Independent Node Discovery
Link Layer Adaptation Layer (LLAL)
(LL-Dependent Node Discovery)
Link Layer 1
LLAL
(LL-Dependent Node Discovery)
Link Layer 2
Link Layer 3

Figure 4.2: Link layer adaptation possibilities

small interface-specific link layer adaptation layer (LLAL) is required. The interface in the middle, however, may not provide such a discovery mechanism and hence its interface-specific adaptation layer needs to implement that. In the rightmost interface, absolutely everything is provided and no adaptation layer is required.

4.3.3 Discovery of Node arrivals and departures

When a Cluster Node detects a new Node, it will try to determine whether there is a Trust Relationship with this new Node. If the new Node shares a Trust Relationship, it is a Personal Node and will immediately be integrated into the Cluster. The Cluster Node will add the information of the new Node to its list of known neighboring Personal Nodes and perform the necessary security mechanisms to establish a secure connection with the new Node. All this is covered in Section 4.4 and Section 4.5.

If the new Personal Node was not already part of the Cluster, more network layer actions must take place before the Node can start to communicate with the rest of the Cluster:

1. Addressing issues might need to be resolved.
2. The routing protocol needs to update its topology.
3. Other auto-configuration mechanisms might need to be performed, such as Gateway Node-detection.

Some of the actions that will take place after the Cluster formation and within the Cluster will be covered in later chapters, in particular:

- Cluster-wide broadcasting (Chapter 5),
- Routing (Chapter 6), and
- Addressing, which also applies to the Cluster organization (Chapter 7).

If the new Node does not share a Trust Relationship, it is a Foreign Node and will never be included in the Cluster. If the Cluster Node is not capable of being a Gateway Node, no connection with the new Node will be established. Otherwise, the Node is capable of becoming a Gateway Node and it adds the Foreign Node to its list of neighboring Foreign Nodes. The Gateway-capable Node can decide to establish a link with the new Foreign Node and thereby become a Gateway Node. However, the Foreign Node will never be able to be included in the Cluster. Instead, the Gateway Node filters out non-authorized traffic from the Foreign Node

It is also important to detect when neighbors disappear. The link layer may report that a Node is no longer connected or otherwise a network layer mechanism based on hello messages can be used to detect Nodes disappearing. When a Node loses the connectivity to a neighboring Personal Node, it will remove that Node from its list of neighboring Personal Nodes. However, this does not necessarily mean that the Node is no longer a member of the Cluster; the Node may still be connected to the Cluster through a different path. It is then up to the intra-Cluster routing protocol to identify a new route. A Gateway Node will also remove a no longer connected Foreign Node from its Foreign Node neighbor list.

4.3.4 Merging and splitting of Clusters

The inclusion of a single Personal Node into a Cluster is a special case of two Clusters merging, since a single isolated Personal Node forms its own one-Node Cluster. When two Clusters with more than one Node merge, the same procedure will take place. The only difference is that the addressing, routing, and auto-configuration protocols may need to do more work, since more than one new Personal Node is being added to the Cluster. The same goes when a Cluster splits up into two Clusters. Only the addressing, routing and auto-configuration protocols may need to do more work.

4.3.5 Cluster member list

In our approach, so far, there is no mechanism that keeps track of the member Nodes of a Cluster. However, other mechanisms, such as the inter-Cluster routing protocol, may need this information. In that case, a table-driven intra-Cluster protocol can be used. If a Node has a route in the routing table to another Node, then that Node belongs to the same Cluster. The routing table will list all Nodes in the Cluster and this information can be used by the Gateway Nodes to construct the right inter-Cluster tunnels and the correct routing over these tunnels. Another option might be to use similar information from the addressing scheme, if available.

4.4 Personal Node Authentication

After a new Node is discovered with one of the techniques mentioned in the previous section, the next question is whether this new Node is a Personal Node or not. This is where the pair-wise keys established during the imprinting procedure come in. Every pair of Personal Nodes shares a Trust Relationship, which means that they have a common pair-wise key, which both Nodes store in their local memory. A cryptographic procedure using the pair-wise keys can determine whether a newly discovered neighbor Node is a Personal Node or not. Figure 4.3 shows a schematic view of the decision process each Node must follow. There are three possible outcomes for each newly discovered neighbor Node:

Personal Node The new Node can successfully authenticate as a Personal Node with the pair-wise key. The new Node is then added to the list of neighboring Personal Nodes and a secure connection using the pair-wise key is established as described earlier. Note that this should be mutual.

Imprinting Node The new Node is being imprinted or wants to be imprinted, i.e., the Node is in the process of being personalized. Exactly which steps will take place here is further defined in Section 3 in [138]. The Node is considered Foreign until the personalization is completed.

Foreign Node This means that the new Node is not a Personal Node or failed to authenticate as a Personal Node.

4.4.1 Neighbor Node authentication

When a new Node is discovered, it can send a message in the clear that claims its association with a certain PN using any unique identifier associated with that PN. A Node that receives such a claim can use that identifier in combination with a unique Personal Node identifier or address to select the correct pair-wise key if it has one. The receiving Node challenges the new neighbor by sending a message with a random number encrypted using the matching pair-wise key. If the new Node can decipher it and send back a different message that includes the same random number and again encrypted using the same pair-wise key, then the neighbor is assumed to be authenticated as a Personal Node. Note that this is a very brief explanation of how a simple neighbor node authentication protocol could work. An actual

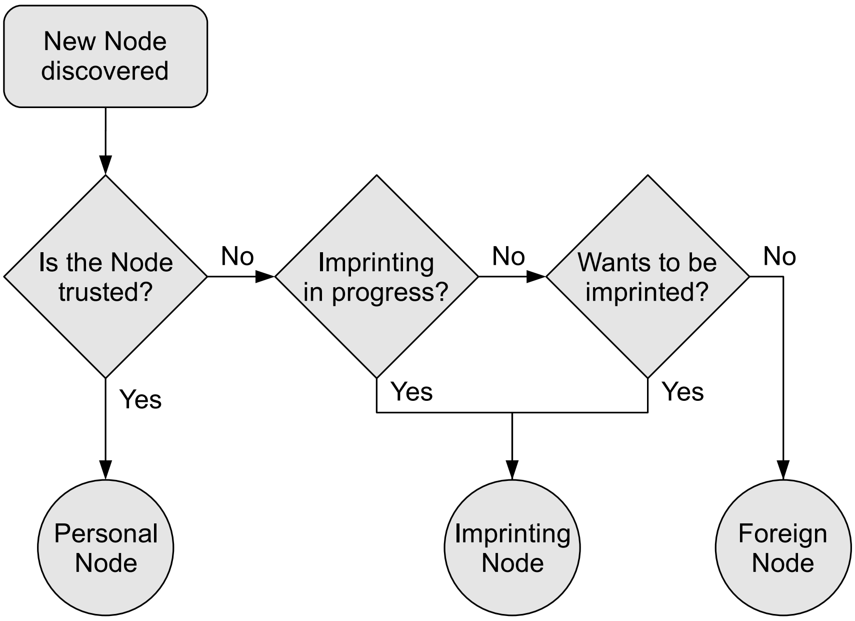


Figure 4.3: The decision procedure for Personal Node detection

implementation needs to include more features in order to fully protect the system against all known attacks.

For this or any other similar schemes to work, it is necessary for the Nodes to transmit their identifiers. This includes both a PN identifier and a Node identifier so that a receiving Node knows exactly which pair-wise key to use. The identifier can be a MAC address, the PN-local IP address or anything else that is fixed. One solution is for each Personal Node to have a mapping between MAC addresses and the pair-wise key for every other Personal Node.

4.4.2 Anonymity

For privacy reasons, it is not always wise to expose fixed and unique identities. Frequently transmitted identities, such as PN and Node identifiers, can be linked to the user of that device and this makes the device and its user no longer anonymous. This also includes fixed network addresses. A mobile device is said to be anonymous if it does not reveal anything that can be used to link it to a person or to a previously encountered device. The latter means that it is impossible for someone to know whether he/she has communicated with or overheard communication from the same device before or not.

When packets with fixed addresses are sent back and forth, it is possible for any third party to know who communicates with whom and at what time. Further, protocol analysis can quite accurately guess what type of data is being sent (e.g., voice, email, or web) even if the traffic is encrypted. For mobile devices, the addresses may also tell something about the location of that device at the time of communication. When all this is put together from many sources, a lot of information can be deduced that may reveal a person's location, who he is communicating with, and at what time—potentially information the person does not want to reveal. Furthermore, anyone can lay their hands on this information, not just network providers.

All this is actually no longer just theoretical threats; a system to track Bluetooth devices was installed in September 2007 in Apeldoorn in the Netherlands [23]. Currently, the system consists of 5 stations at different locations recording anyone passing by with a device with Bluetooth switched on. If you know someone's Bluetooth MAC address, you can detect when and if that device has been seen on any of the locations through a website [23].

Hence, it is necessary to hide such identities to remain anonymous. Unfortunately, that also makes it harder for the Nodes to select the correct pair-wise key when encountering new neighbors. In the worst case, all stored pair-wise keys must be tested one by one before the authentication fails and the Node is declared Foreign. This is, of course, a waste of communication and computational resources and better solutions are needed.

One very simple idea, we propose in lack of better options, is to have one PN-wide key that is shared by all Nodes in a PN. The key can be exchanged during the personalization so that every Node in the PN can store it in its local memory. The key should only be used to encrypt PN and Node identities and nothing else. Then, breaking of this key only leads to lost anonymity, but not to access to the PN network. The latter still requires breaking the pair-wise key.

While it is possible to encrypt all fixed addresses and identifiers in the network layers and higher, it is also necessary to protect fixed addresses used by the lower layers. Most wireless technologies use unique identifiers that can be read and used in clear and it is simply not possible to encrypt them, since they are used to determine the sender and the receiver. WLAN (IEEE 802.11) is one example. Each WLAN card has a hardware address of six bytes that is used when communicating with another WLAN-equipped device or access point. To make sure that addresses never clash, each WLAN card is equipped with a globally unique address. This address can, of course, be used to map to the owner of the laptop it belongs to. Furthermore, the WLAN card is constantly transmitting its hardware address in clear, even when encryption is used. This is still the case when using the latest WLAN encryption standards, such as Wireless Protect Access (WPA) and WPA2 [83]. Also Bluetooth suffers from the same problem since it uses the same type of hardware addresses. However, Bluetooth is slightly better as it does not transmit its unique hardware address with every message. It is only sent as part of the piconet formation procedure after discovering a new neighboring device.

The best answer to this problem is to get rid of the uniqueness and frequently change link layer addresses, since global uniqueness is not really needed anyway. The address only needs to be unique among the devices in the close vicinity, where its transmissions can be received. Even if there are address conflicts, they can be solved by inspecting the (encrypted) network layer address or by the failure to decrypt the packets. A conflict only degrade the performance somewhat. The addresses can easily be manipulated anyway and hence cannot offer any additional security.

We should also note that just as there is no perfect security, perfect anonymity is impossible. There will always be information leakage that may be used to guess who the user of a device is. However, this must not stop us from developing techniques that make this more difficult. Instead, the most obvious vulnerabilities must be avoided which will make it much harder for an adversary to discover this kind of information. The first step is to avoid fixed addresses that are sent unencrypted and can be overheard by non-authorized peers. When that is taken care of, an attacker needs to turn to the physical layer to look for clues. One example is using a technique called radio frequency fingerprinting (RFF) [70]. However, techniques like this one are also much more difficult than just listening for unencrypted fixed identifiers.

4.5 Establishment of Secure Communication

When a neighbor Node has successfully been authenticated as a Personal Node, it is necessary to establish a secure communication link for both unicast and broadcast traffic. Also the link layer mechanisms, such as piconet formation, etc., should be protected if possible.

4.5.1 Secure unicast communication

The pair-wise key can be used to derive or securely exchange a link layer session key for data encryption and integrity protection. How the link layer session key is established depends on the link layer technology. If there is no data encryption and integrity protection provided by the link layer, this has to be implemented at the network layer instead.

The pair-wise key should not itself be used to encrypt all the data traffic. It is a key that should not need to be changed too often, which means that its use should be minimized. Hence, the pair-wise key is only used to establish session keys and for Personal Node authentication. The session keys are used to encrypted the data traffic between two neighboring Personal Nodes.

The various link layer technologies should use the session keys to encrypt and decrypt the packets at the link layer whenever the link layer provides adequate security mechanisms. Hardware-based encryption provided by the network interface cards can then be used and this improves both performance and power consumption. For an end-to-end path between two Personal Nodes

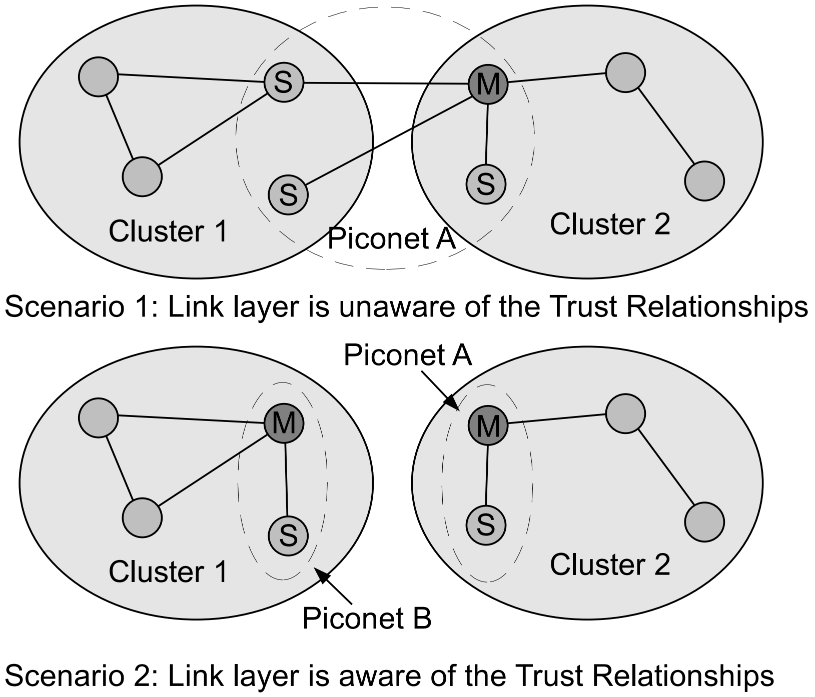

S
M
S
S
Cluster 1
Piconet A
Cluster 2
Scenario 1: Link layer is unaware of the Trust Relationships
Piconet A
M
M
S
S
Cluster 1
Cluster 2
Piconet B
Scenario 2: Link layer is aware of the Trust Relationships

Figure 4.4: Link layer aware or non-aware of Trust Relationships

within the same Cluster, this encryption is performed on every hop and this is the way intra-Cluster unicast traffic is protected. This method is fully ad hoc and fully distributed as well as secure if adequate encryption is used on each hop. This has been developed further by IST MAGNET and is reported in [138].

4.5.2 Trust Relationship-awareness at the connectivity level

Personal Networks and Cluster formation must work on link layer technologies that are Trust Relationship agnostic. At the same time, many link layer technologies form piconets or other types of logical structures. They do that to better mitigate collisions and contention on a shared wireless channel or to enable power saving mode. However, this may lead to some unwanted situations. Consider the two scenarios shown in Figure 4.4. The figure shows the Clusters of two different persons, which means that they must never merge. The first scenario is what may happen if Bluetooth is used as one of the link layer technologies (other link layer technologies may produce a similar result). The Bluetooth link layer is unaware of the Trust Relationships between Nodes and may easily form one single piconet consisting of Nodes from both Personal Networks. A piconet controller may control the medium access of Nodes of another Cluster.

Future link layer technologies may of course be specifically designed for Personal Networks and hence understand Trust Relationships and Cluster formation requirements. In the second scenario, the piconet formation protocol is aware and is forced to include only Personal Nodes. The second scenario provides better security, which may tempt us to suggest such a solution despite that it requires changes to the link layer. However, the first scenario provides better medium access management possibilities, since it works across the Cluster boundaries.

The choice whether the link layer should be Trust Relationship-aware is all but clear. In many cases, it is desirable to be protected against radio jamming and other types of denial-of-service (DoS) attacks. However, when the contention increases due to many devices sharing the same wireless channel, then contention and collision mitigation schemes operating between the involved Clusters would be valuable. The situation can be quite bad when lots of people gather in large crowds, such as sports events, large expositions, and the like. The ultimate system should coordinate medium access and at the same time protect the system against DoS attacks.

4.5.3 Secure broadcast communication

Not only secure unicast communication is important, secure Cluster-wide broadcasting is also required. Several mechanisms require it, such as service discovery and dissemination of context information. To implement a secure Cluster-wide broadcasting mechanism, we need secure broadcasting at the link layer. Link layer broadcasting is typically only one hop, but will be an important building block for Cluster-wide broadcasting. In this section, we discuss how the link layer can provide this building block in a secure way. Chapter 5 covers the topic of Cluster-wide broadcasting using flooding at the network level.

Clusters consist of several different link layer technologies and each of them implements its own way to secure link layer broadcasting, if they implement a secure broadcasting at all. Even if a current link layer technology is capable of secure link layer broadcasting, it is anyhow unaware of Trust Relationships. Based on this, we see three possible solutions to the Cluster-wide broadcasting security problem:

1. We could use a unicast-only flooding protocol (e.g., [121]) for Cluster-wide broadcasting and completely avoid everything about broadcast keys and link layer broadcasting. This adds reliability to the broadcasting since we can use ACKs, RTS/CTS, etc. However, it is less efficient in terms of number of transmitted messages, especially since Clusters are compact and a broadcast medium is used.

2. Distribute a Cluster-wide broadcast key in the whole Cluster and use that key for all broadcast traffic. Unfortunately, distributing that key is

a Cluster-wide broadcasting problem by itself. The only option would be to distribute the key using a unicast-only flooding technique as above. Furthermore, when two Clusters with different broadcast keys merge, they have to converge, which makes this proposal extra complex. Since we depend on the security features of the link layer technologies, there are almost no benefits of this solution.

3. Each Personal Node has its own link layer broadcast key to broadcast to the immediate Personal Node neighbors. If the link layer does not provide secure link layer broadcasting, it can be implemented in higher layers. The broadcast keys can be exchanged with the unicast session keys (with a unicast message) right after two Personal Nodes find each other in such a way that only Personal Nodes have access to the keys.

The third option offers the best solution since it keeps the broadcast communication to a minimum and at the same time it keeps the amount computation low. Each message needs to be decrypted at each hop in the first place and decryption/encryption can take place in the hardware using the link layer solution. Node mobility within a Cluster is also less of a problem in this solution. The broadcast key is exchanged as part of the standard neighbor discovery process and Nodes may keep the broadcast key even after they lost contact with a neighbor. In the case that a broadcast message is received from a disappeared neighbor, it will still be able to decrypt it unless the broadcast key has been updated, which does not need to happen often.

4.6 The Prototype

To better understand Cluster formation and fill eventual gaps in the protocol specification, we implemented a Cluster formation prototype. The prototype is based on the Linux operating system and partly builds on available software. This section introduces the prototype implementation and our findings.

4.6.1 Hardware platform

The prototyping was done on six standard laptops. This allowed us to use the standard Linux operating system without any non-standard tailoring, which is typical when using smaller devices. The six laptops were equipped with Ethernet and WLAN based on IEEE 802.11 [82]. Two laptops had an Intel Core 2 Duo 1.66 GHz processor, two had an Intel Celeron M 1.6 GHz processor, and two had an Intel Mobile Pentium 4 at 3 GHz processor. All laptops had 512 MB RAM. For the wireless, we mainly relied on 3Com OfficeConnect Wireless 108Mbps 11g XJACK PCCard [1], but also on the built-in Intel PRO/Wireless 2200BG WLAN card available in some of the laptops. The support under Linux of these wireless interfaces was good.

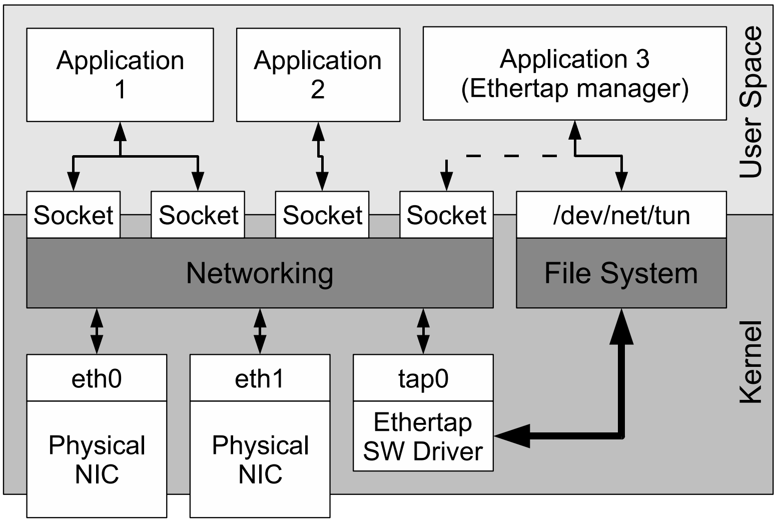


Figure 4.5: Ethertap architecture

4.6.2 Linux

Linux [120] is a free open-source operating system. Linux runs on many different hardware platforms, including PCs, Laptops, PDAs, etc. It provides everything that can be expected from a state-of-the-art operating system. Since the source code is available, it is possible to adapt the system in the way we need to build a Cluster Node prototype. We need to make modifications to the network layer subsystems, including the routing implementations, which is not possible in most commercial operating systems. Linux is therefore the natural candidate operating system for a prototype.

Linux provides a virtual network interface in software called Ethertap [58]. To the rest of the system, Ethertap (called `tap0` if not given another name) looks like a normal Ethernet interface except that it does not physically exist as shown in Figure 4.5. Instead, everything that is sent out through the interface is passed on to a specially connected program that manages the virtual Ethertap interface. Through a special device file (`/dev/net/tun`), this program can access packets sent onto the Ethertap interface, process the packets, and possibly reply to the system by passing packets back to the Ethertap interface via the same device file. Such packets are passed to the interface as if they were received on a physical interface. In this way, Ethertap can emulate real network traffic.

An Ethertap interface can be used to implement a virtual interface for intra-Cluster communication. A user application that wishes to send intra-Cluster data has to send it through this virtual interface; this is done by configuring the kernel routing table. The connected program receives the traffic via the virtual interface and takes the appropriate actions, such as encrypting the packet, and forwarding it to one of the real interfaces for actual transmission to a neighboring Personal Node. In this way, no modifications are needed to the kernel or the user applications.

4.6.3 OLSRd

To fully prototype a working Cluster, we also need addressing and routing. The addressing was done manually, but could be done automatically as part of the personalization functionality. A special prefix (temporarily 3000::/16) was assigned for all intra-PN addresses. For the routing, we use OLSRd [169], which is a Linux implementation of the optimized link state routing protocol (OLSR). The idea is to use OLSR as the intra-Cluster routing protocol. Since Ethertap emulates a real network interface, OLSRd can operate on top of the virtual interface with no modifications. If the Ethertap interface, through the specially connected program, filters out Foreign Nodes, OLSRd will only know the Personal Nodes and only route between them, i.e., OLSRd will be doing intra-Cluster routing without any modifications. While there is no absolute need to modify OLSRd, some modifications are advisable. Most notable is the way OLSRd detects one-hop neighbors. OLSR implements its own hello protocol, which of course is not necessary. A better way would be if the Cluster formation functionality feeds this information to the routing daemon instead. This requires some modifications to the daemon and for OLSRd, those modifications are significant. In fact, the routing protocol itself needs to change due to OLSR's way of integrating the routing with the hello protocol.

4.6.4 Implementation architecture

With the implementation prototype that we developed, no changes are required to the applications; they use the normal way of sending and receiving traffic. The intra-Cluster network layer is based on IPv6 [48] and currently supported network interface types are fixed Ethernet and WLAN according to the IEEE802.11 family. Node discovery is implemented at the network layer and is based on UDP packets and link local multicasting typical for IPv6. The various packet formats are shown in Figure 4.6.

The link layer security features of the supported network types are disabled. Instead, we show the feasibility of implementing the intra-Cluster communication mechanisms entirely at the network level for both fixed Ethernet and WLAN. Data packets are encapsulated in UDP and sent with IPv6 packets using link local addresses as shown in Figure 4.6. Obviously, it is unnecessary to encapsulate data packets like this, it would be better to introduce a new network protocol that only has a short header and whose purpose is to encrypt the entire intra-Cluster communication packet. However, we made this choice since it makes it easier to implement. The intra-Cluster formation

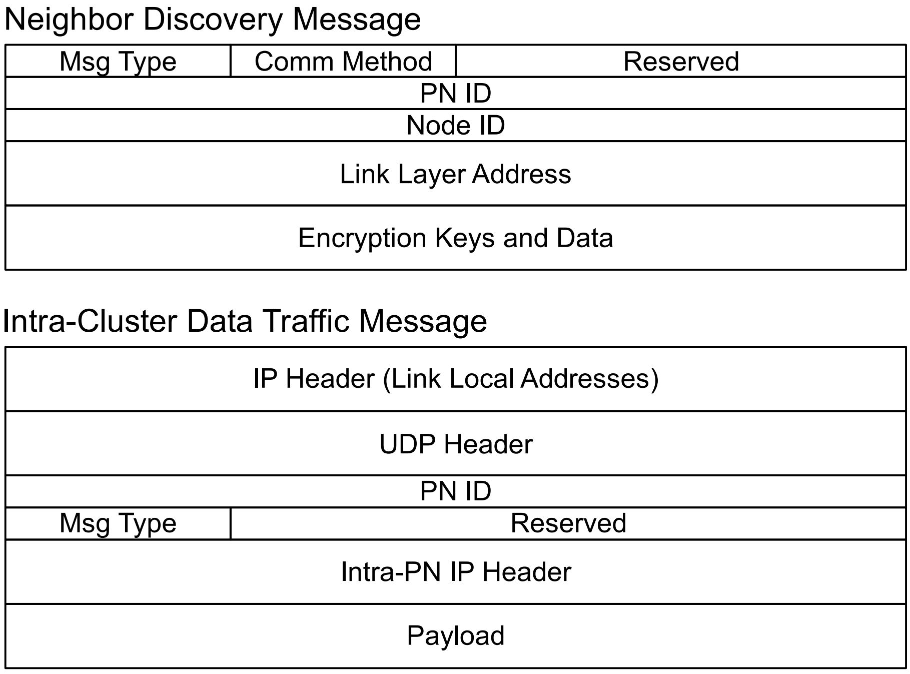


Figure 4.6: The intra-Cluster prototype packet formats

and communication program can then receive and send packets in the normal way.

Figure 4.7 shows a schematic view of the prototype implementation of a Node without Gateway Node functionality. Thick arrows denote data traffic and thin arrows denote control, routing, and device discovery traffic. The virtual interface implemented by Ethertap is called `ppan1`. It is the interface always used for intra-Cluster communication. The program connected to `ppan1` is called ppand. It is also ppand that implements the Personal Node discovery and authentication process. It maintains the Personal Node neighbor table (PNNT) and makes sure packets on the virtual intra-Cluster interface (`ppan1`) are encrypted before being sent to a neighbor. Security functions are not yet implemented in the prototype due to lack of time. Proper Personal Node authentication and hop-by-hop encryption require some extra message exchanges and headers, but otherwise, the implementation will be similar.

OLSRd is configured to send and receive its routing packets only over the virtual intra-Cluster interface. OLSRd directly updates the kernel routing table. Each destination within the Cluster is pointing to the virtual intra-Cluster interface, but with different next hop addresses. The discovery or disappearing of neighbor Nodes will add or remove visible Nodes from the virtual intra-Cluster interface. As soon as this happens, the routing daemon can send and receive hello messages to and from these neighbors and detect topology changes. Also broadcast or multicast packets sent to the virtual intra-Cluster interface are only received by neighboring Personal Nodes.

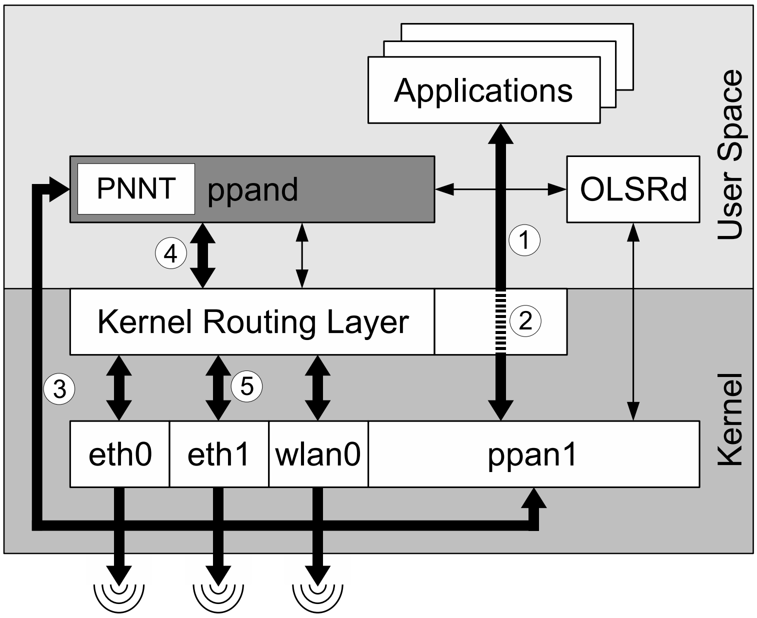


Figure 4.7: The Cluster prototype implementation architecture

However, since ppand implements a Personal Node discovery protocol, it is unnecessary for OLSRd to have its own. It is better if ppand informs the routing daemon directly when a Node is discovered or disappears. Therefore, a special IPC mechanism was designed to connect ppand and the routing daemon.

It is also worth pointing out that data traffic actually passes through the kernel routing module twice. First, all intra-Cluster traffic is passed to the virtual intra-Cluster interface and then, ppand adds a second IPv6 header with a link local address of a directly neighboring Personal Node. Before the packet is sent to the next hop, it passes the routing table once more.

4.6.5 Sending intra-Cluster traffic

Let us go step by step through the process of sending a packet. Consider an application on Node 3000::1 in Figure 4.8 sending a packet to 3000::3. Following the steps in Figure 4.7, this is what takes place:

1. The application sends the packet to an intra-Cluster address (3000::3). The packet will be delivered to the kernel through a `send()` call and the routing layer in the IP stack will be called.

2. The routing layer will find the next hop IP address, which is also an intra-Cluster IP address. The next hop IP address is bound to the virtual intra-Cluster interface (`ppan1`), but before the routing layer can send it, it needs the link local address of 3000::2 on `ppan1`.

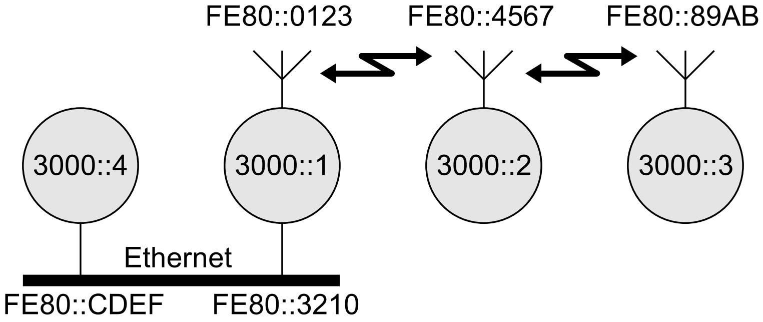


Figure 4.8: An example scenario

Table 4.1: Personal Neighbor Node Table of 3000::1

Personal Node Address	*Current Link Local Address*	*Interface*	*Pair-wise key*
3000::2	FE80::4567	`wlan0`	90ab74bf582b3e28c
3000::4	FE80::CDEF	`eth0`	a8493eb57e7f43cc8

3. The kernel will then try to find the link layer address (Ethernet address when using Ethertap) for 3000::2 on `ppan1`. A Neighbor discovery packet (ICMPv6) is first sent to `ppan1`. Here, ppand answers this request with a unique address. Any address will do as long as it is unique on `ppan1` and always the same for 3000::2. The kernel will then send the packet using that link layer address as link layer destination and 3000::3 as the IPv6 destination. When ppand receives this packet, it knows both the next hop (from the link layer address) and the final destination (from the IPv6 address). This process taking place between the routing layer and ppand can be seen as a pure internal mechanism and can be optimized in a real implementation.

4. The ppand has a Personal Node neighbor table (PNNT), which is shown in Table 4.1. PNNT should contain the next hop Node. Ppand will use the pair-wise key of the next hop (3000::2) to encrypt the packet. It also extracts the link local address (FE80::4567) and interface (`wlan0`) of the next hop Node and sends it to the kernel using `sendto()`.

5. The kernel believes the packet is for the next hop, so it forwards it to the correct real interface for transmission as usual.

Table 4.1 shows the PNNT of Node 3000::1 in this scenario. This table contains only Personal Neighbor Nodes and maps PN-internal addresses to external addresses. In this prototype, we use the IPv6 link local address of the Node. In a real implementation, the MAC address is a better option. We also need to keep the pair-wise key for encrypting/decrypting the data traffic to and from the neighbor node.

4.6.6 Interface output queue

It is necessary to give control packets, such as hello packets and routing packets, higher priority when queue arises on the intra-Cluster interfaces. This is done on the output queue of each of the interfaces using the Linux traffic control feature [9]. The traffic control decides which packet to send first, whether to delay a packet transmission, which packets should be dropped, etc. The default output queue on Linux is a first in first out (FIFO) queue with tail drop. This is changed by a configuration tool called `tc`, available in the iproute2 package. With `tc`, advanced queuing structures can be set up for each outgoing interface.

To make sure hello packets and routing packets are never delayed or dropped due to full buffers, we introduced a priority queue on each interface used for intra-Cluster communication. The priority queue had only two priorities; a high one, which is always processed first, and a low one. Filters were installed that made sure hello packets and routing packets got the higher priority. Since both routing and data packets are encrypted, we simply gave all broadcast packets higher priority since all routing traffic uses broadcast. While the high priority queue had a length of 10 packets, the low priority queue had a length of only 2 packets. This was done to avoid stale data packets being queued everywhere in our experiments.

4.6.7 Receiving intra-Cluster traffic

Ppand binds to the UDP port used for intra-Cluster traffic and receives all intra-Cluster packets for processing. It looks at the source address and identifies the neighboring Personal Node and its pair-wise key in the PNNT. Ppand then tries to decrypt the content and if successful, sends the decrypted packet to `ppan1` for further processing. The kernel routing table decides whether the packet is to be sent to one of its applications or whether to forward it to another Node. In the latter case, the packet is forwarded back to `ppan1` and ppand for encryption and transmission to the next hop.

4.6.8 Lessons learned

No major problems were encountered during the implementation. The prototype worked smoothly on the laptops we used. The code size is small with only 6500 lines of code (LoC) in the C programming language which also includes the code of the functionality used in Chapter 5 and Chapter 6. Only the security implementation is missing. Hence, we see no major problem on running this on embedded devices. If a device can support IP, then it can support Cluster formation and communication.

We used Ethertap to be able to develop the code in user space instead of kernel space. This was done for simplicity. It made it easier to implement, but more importantly, much easier to test and debug. Because of this, data packets are passed between user and kernel space several times and one extra time through the kernel routing table. Despite all this, no real performance degradation could be noticed compared to if the two laptops would communicate without the PN prototype with Ethertap and all.

All applications we tried worked without modifications as long as they support IPv6. This included SSH (secure remote shell), SCP (secure file transfer), Firefox (web browser), and a VoIP Application. Furthermore, we found that a Cluster could become multi-hop without problems when we enabled OLSRd.

A problem discovered during the prototyping was the handling of the maximum transmission unit (MTU). We had to assign a MTU for the virtual Ethertap interface as well. Normally, the MTU specifies the maximum payload allowed by a link layer. However, the virtual interface can dynamically select the outgoing interface and the different interfaces can have different MTUs. Hence, it is not clear what the MTU of the virtual interface should be. The best option is probably to assign the MTU to the interface with the largest MTU minus the Cluster-specific headers. However, this may trick the kernel to generate too big packets when another link is chosen and this creates extra signaling in the form of ICMPv6 Packet Too Big messages. However, this is actually quite normal and the kernel or the source Node reduces the packet size when such a signaling message is received.

4.7 Summary

In this chapter, we studied Cluster formation and maintenance. Clusters are nothing more than active Personal Nodes that can communicate with each other using their own communication features and without external support. Clusters are basically ad hoc networks and may therefore consist of multiple heterogeneous wireless technologies and hence be multi-hop networks.

We defined a Cluster as connected network consisting of active Personal Nodes exclusively and argued that this is the best one. It makes intra-PN communication rely on local communication to the largest possible degree. Since we believe local communication will outperform long distance communication most of the time, this should lead to the most optimal solution.

Then we walked through all the necessary mechanisms for Cluster formation and maintenance, such as neighbor discovery, Personal Node authentication, anonymity, and establishment of secure intra-Cluster communication. In the last section, we presented how we implemented a Cluster formation prototype on laptops running Linux. Details were given on how it worked and how it performed. Measurements indicate that only encryption may add any significant overhead to the intra-Cluster communication. Hence, we believe that we demonstrated the feasibility of our Cluster formation and maintenance approach.

Chapter 5

Cluster-Wide Broadcasting

PNs need Cluster-wide broadcasting. For many applications, it is necessary to transmit a packet from one Node to all the others inside a Cluster. However, in a multi-hop Cluster, the range of the source Node does not cover the whole Cluster and therefore needs help from other Nodes to relay the packet to reach all Nodes. This process, we refer to as *flooding*, which is an important feature for any wireless multi-hop network.

Flooding is used by several unicast routing protocols [179][104] for multi-hop networks to disseminate route requests or link states. Other applications include service discovery, sharing of context information, address autoconfiguration [30], and network self-organization. In mobile networks, flooding is often a better choice than multicasting due to the frequent topology updates [167]. Understanding flooding will help us understand other aspects of multi-hop networks as well.

The simplest flooding mechanism for multi-hop networks is Blind Flooding, in which a node always retransmits a received flooding packet after a small random delay (jitter). Each node needs to keep a list of recently received flooding packets to detect duplicates and avoid retransmitting the same packet twice. However, this is the only mechanism in Blind Flooding that reduces the number of retransmissions.

Several studies of flooding protocols have revealed that it is possible to reduce the number of retransmissions much more [225][231]. This has the benefit of reducing contention, collisions, saving energy, and it may even increase the flooding speed. Such optimized flooding protocols can make use of location information, neighbor information, or observations of the ongoing flooding. Protocols using location information require solutions like global positioning system (GPS) in every node, which is too strong a requirement for most PN devices. Besides, the improvements brought by location information have not proven to be very significant [231], especially if there is mobility. We therefore assume in our research that the nodes do not have such information.

This chapter is structured as follows. Section 5.1 introduces the most important requirements for Cluster-wide broadcasting. Section 5.2 lists related work, including existing flooding protocols suitable for Clusters and previous experiments with flooding. In Section 5.3, we propose a new flooding protocol, called Prioritized Flooding with Self-Pruning (PFS) [99], which is an optimization and combination of two existing flooding protocols. We then verify PFS's performance by a series of measurements in a real wireless multi-hop network, which is introduced in Section 5.4. The experimental network is intended to give us a better understanding of how the flooding protocol behaves. Typical Clusters will look quite different, but the results we obtain using our experimental networks are also directly applicable to any type of Clusters. In Section 5.5, we present and discuss the results of the measurements. We attempt to simulate PFS' performance in Section 5.6, but conclude that the simple network model used in ns-2 [159] is insufficient for our purposes. In Section 5.7, we introduce how we enhanced our prototype from Chapter 4 to also do optimized flooding within a Cluster. Section 5.8 summarizes this chapter.

5.1 Cluster-Wide Broadcasting Requirements

To achieve satisfactory Cluster-wide broadcasting, we need a flooding solution that fulfills the following requirements:

1. Support for heterogeneous link layer technologies. This is an important requirement, since Clusters can be connected using various link layer technologies simultaneously.

2. The solution must be able to deliver a Cluster-wide broadcasting packet to all Nodes in the Cluster with a very high probability. A Cluster is per definition connected and that means that a flooding protocol must be able to reach all Nodes. Applications should be able to depend on the flooding reaching all Nodes in the Cluster.

3. The amount of resources used, such as the number of transmissions, packet overhead, and computation, should be minimized. Most importantly, unnecessary transmissions should be avoided. This minimizes collisions, contention, and the impact on other traffic.

4. No unnecessary delays should be introduced. Many applications require Cluster-wide broadcasting to be timely.

5. Node mobility within the Cluster must be supported. Since Cluster Nodes are mobile, mobility must be handled without jeopardizing the other requirements.

6. Robustness against errors and security attacks. External as well as internal disturbances, intentionally or unintentionally ones, should be

dealt with in an appropriate way while still satisfying the other requirements.

To meet all requirements is very difficult. Many of them are even contradictory. This means that a balance must be achieved. Unfortunately, what is sufficient and what the balance should be is extremely difficult to quantify. In this chapter, we will mainly focus on requirements 2 to 5.

5.2 Related Work

Any flooding protocol for general ad hoc networks is, in principle, applicable also to Clusters, as long as it supports the requirements above. Many different flooding protocols have been proposed in the past. In this section, we only introduce the most common protocols and protocols similar to PFS. For a more extensive survey of flooding protocols, see [231][215]. At the end of this section, we comment on previous experiments with flooding in real networks.

5.2.1 Counter-Based Broadcasting

The first flooding protocol we discuss is Counter-Based Broadcasting (CBB). The authors of [225] found that as the number of times a node hears its neighbors retransmitting the same flooding message, the chance that one of its direct neighbors has not yet received that message quickly becomes very low. This can be explained by looking at the area that a transmission covers. The expected additional coverage of a node dramatically decreases each time it hears the same message. The authors of [225] observed that after 4 times, the expected additional coverage is below 5 % of the node's entire coverage area if we assume perfect circular coverage areas. This means that the probability of covering an extra node is very low.

CBB works as follows. When receiving a previously unheard flooding message, a node waits for a random assessment delay (RAD), which is uniformly chosen between 0 and T_{max} seconds. During the RAD, the node keeps a counter that counts the number of times it receives the same message. When the RAD expires, the node retransmits the message unless the counter has reached a threshold k. The theoretical results in [225] suggest that k should be 4 or less.

However, CBB does not ensure reachability even under perfect channel conditions, since the nodes are not necessarily evenly distributed and there is always a chance that a node that must retransmit refrains from retransmission because it exceeds the threshold. Increasing k gives a better reachability, but also reduces the benefit of CBB. In practice, CBB achieves very good reachability in many situations as we will demonstrate later in this chapter.

5.2.2 Flooding with Self-Pruning

Flooding with Self-Pruning (FSP) [119] is a simple protocol that belongs to the class of neighbor knowledge-based protocols. Protocols in this class gather information about their neighbors by exchanging hello messages on a regular basis. FSP is based on self-pruning (or neighbor elimination) and uses one-hop hello messages (each node knows its one-hop neighbors using hello messages). Every node encloses its neighbor list in the flooding message header and by this mechanism, a node that receives a flooding message knows which other nodes are "covered" by that transmission. A node checks whether all its neighbors are covered by the sender by comparing the sender's neighbor list and its own neighbor list. If it still has "uncovered" neighbors, it retransmits the message; otherwise, it does nothing.

Although there are no simulation or measurement results in the literature for FSP so far, we can imagine that FSP does not reduce retransmissions a lot compared to Blind Flooding. Only nodes that are very close to the sender may have all their neighbors covered by the sender and refrain from retransmitting.

5.2.3 Scalable Broadcasting Algorithm

Scalable Broadcasting Algorithm (SBA) [173] is also a protocol based on self-pruning, but requires two-hop hello messages. Each hello message contains the neighbor list of the sending node. Hence, each node knows their neighbors as well as their neighbors' neighbors. Upon receiving a new flooding message, the node compares the sender's neighbors with its own neighbors. If not all of its neighbors are covered and it hence can cover extra nodes, it schedules a retransmission of the message after an RAD. The main difference with FSP is that if, during the RAD, duplicate messages are received which together cover all neighbors, the retransmission is canceled. In this way, SBA expects to achieve good performance.

To achieve even better performance, the T_{max} of the RAD depends on how many neighbors the node has in comparison with the number of neighbors its own neighbors have. Each node calculates its T_{max} by dividing the highest number of neighbors among its neighbors by its own number of neighbors including themselves and then multiply with a scaling constant. For node A, that is:

$$T_{max} = C \cdot \frac{1 + \max_{b \in N(A)} |N(b)|}{1 + |N(A)|}$$

where $N(x)$ is the neighbor set of a node x and C is the scaling constant. The effect of this is that nodes with more neighbors are more likely to retransmit faster and this should make the self-pruning more efficient. Unfortunately, the improvement is limited, which we showed by simulations in [99].

5.2.4 Ad Hoc Broadcasting Protocol

Ad Hoc Broadcast Protocol (AHBP) [174] also requires two-hop hello messages. Unlike FSP and SBA, it is the sender that decides which of its neighbors should retransmit the flooding message. This decision is based on the two-hop neighbor information that the node has. The selected neighbors are called Broadcast Relay Gateways (BRGs) and are listed in the header of the flooding message.

The BRGs are selected in such a way that if they retransmit the flooding message, they will together cover all two-hop neighbors of the sender. The receiving nodes repeat this procedure and this guarantees that all connected nodes in the network will receive the flooding message as long as the two-hop neighbor information is accurate. To decide which neighbors are BRGs, the following greedy algorithm is used:

1. Set the BRG list to empty and find all two-hop neighbors that have not yet received the flooding message.

2. Identify uncovered two-hop neighbors that can only be covered by one single one-hop neighbor. Include all these one-hop neighbors in the BRG list. Remove all two-hop neighbors that are covered by these one-hop neighbors.

3. Find the one-hop neighbor that covers most uncovered two-hop neighbors. Add this node to the BRG list and remove the two-hop neighbors it covers.

4. Repeat step 3 until all two-hop neighbors are covered.

Previous research [174] [231] has shown by simulation that AHBP performs very well in static networks. However, when the mobility increases in the network, the two-hop neighbor information becomes inaccurate and the reachability decreases. Increasing the frequency of hello messages improves the performance but also adds a lot of overhead. To better cope with outdated neighbor information, the authors propose an extension to AHBP. This extension tells a node that receives a flooding message from an unknown neighbor (it has not yet received a hello message from this neighbor) to assign itself as a BRG and retransmit this message. This extension increases the reachability of AHBP in case of mobility and is an important extension to AHBP. However, this extension only reduces the effect of outdated neighbor information, it does not make it disappear.

5.2.5 Other flooding protocols

Many other flooding protocols have been proposed throughout the literature; too many to mention them all here. Examples include Multipoint Relays

(MPR) [7], Essential Connecting Dominating Set (E-CDS) [168], and many more (e.g., [42]). MPR is an important one, since it is used in the OLSR routing protocol [37]. MPR is a similar to AHBP, but the BRGs (called MPRs in [37]) are communicated with the neighbors in the hello messages instead of the flooding messages.

The IETF MANET working group [143] has also started standardization effort targeted for flooding in multi-hop networks [128]. This work is just a framework for flooding, leaving the choice of flooding protocol open. That is, the framework only specifies mechanisms related to packet formats, neighborhood information collection, and how nodes should detect that a received packet is a duplicate. Several example flooding protocols are given in the document and how they could be implemented using the framework.

Also in the area of wireless sensor networks (WSN), flooding has been considered. The main application in WSNs for flooding is to disseminate code updates and instructions from a single sink to all the sensor motes in the network. Hence, the assumptions and requirements are different. Most WSN flooding protocols assume a static network with only one single flooding source. Further, concurrent traffic is assumed to be insignificant and reachability is the most important criteria. Since code updates is the most important application, they also assume a series of associated flooding messages with which lost packets at a mote can be detected by missing packets. Examples of WSN flooding protocols are Deluge [78], Pump Slowly and Fetch Quickly (PSFQ) [228], Sprinkle [156], and Trajectory- and Energy-Based Data Dissemination (TEDD) [127].

5.2.6 Real networks experiments

Not many studies of flooding in real networks have been performed. So far, only a few experiments are known and most of them studied Blind Flooding or a variant thereof.

The authors of [61] used 150 Rene motes in a 12 x 13 grid network to study the behavior of Blind Flooding. The authors observed that instead of a step by step rippling outward, a flooding sometimes extends backwards and towards the source. Furthermore, some close neighbors can not receive the flooding message, while some distant nodes can hear the same retransmission through a longer link than expected. All these effects showed that the behavior of a simple flooding protocol is surprisingly complex in realistic experiments.

In [220], the author evaluated Blind Flooding on 10 HP iPAQ 5500 PDAs with integrated IEEE 802.11b radios. In the first experiment, 7 PDAs were placed in a parking area of about 165m x 90m. They found that the reception rate was quite low (under 50 %), which was explained by poor connectivity. In a second experiment, 10 nodes were placed in a denser network with a size of 108m x 86m. Due to the denser network, the reception rate was much higher (more than 90 %). The author also experienced links with great packet loss asymmetry and changes over time.

In [224], the authors deployed a dense network of 88 nodes in an office environment to test the performance of the Drip protocol. Drip is a modification of Blind Flooding for disseminating commands in a wireless sensor network running TinyOS. To deliver a flooding message reliably, Drip continuously retransmits a sent message after waiting double the amount of time than last retransmission. Drip was compared with Blind Flooding. Both protocols achieved 100 % reachability in the experiments.

Also for WSNs, flooding experiments have been done. In [78], Deluge was tested in a WSN of 75 motes in an office environment and proven to work in practice. Further, the authors of [156] used the ExScal test bed [13] to test the performance of their flooding protocol Sprinkle and to compare it to Deluge. In ExScal, 203 nodes equipped with IEEE 802.11b radios and GPS-receivers were used. The conclusion of the ExScal experiment was that Sprinkle significantly reduces retransmissions and delay compared to Deluge. However, Sprinkle requires node locations and a static network.

5.3 Prioritized Flooding with Self-Pruning

In this section, we introduce Prioritized Flooding with Self-Pruning (PFS). The original protocol was introduced in [99], but when tested in real wireless multi-hop networks, we experienced some issues. In general, the original protocol works in real networks. However, under some circumstances, due to effects not found in the perfect world of simulations, the number of retransmissions could significantly increase. We therefore adapted the protocol with some changes to make it more fit for real networks. This section will explain PFS in its entirety, including the changes we made, such as the limit to the number of slots and the early retransmission function.

5.3.1 Self-Pruning Aspects

PFS is similar to SBA. It uses self-pruning and RAD for scheduling of retransmissions. However, PFS only uses one-hop hello messages and instead includes the neighbor list of the sender in the flooding message, which is similar to FSP.

One-hop hello messages are much smaller than two-hop hello messages. Hence, the overhead brought by hello messages can be reduced. Furthermore, one-hop hello messages are often already required by other mechanisms in the network layer (e.g., by routing protocols) or already provided by the link layer protocol (e.g., device discovery).

Each node transmits its neighbor list in each flooding message. When there is a lot of flooding traffic in the network, this may lead to increased overhead. If at the same time the mobility is low, the same neighbor list is enclosed in several consecutive flooding packets, which is unnecessary. A possible extension is to let a node only send its neighbor list when something has changed. This was investigated in [99] and shown to provide some improvements.

```
For a node B
On receiving a flooding message $m$ from $A$
   If message $m$ received for first time
      $C(m) = (N(B) \cup B) \setminus (N(A) \cup A)$
      $d(m) = \mathrm{RAD}_{\mathrm{PFS}}(|N(A)|, |C(m)|, |m|)$
      Delay $m$ for $d(m)$ seconds
   Else
      $C(m) = C(m) \setminus (N(A) \cup A)$

When the delay $d(m)$ expires
   If $|C(m)| > 0$
      retransmit message $m$
   Else
      drop message $m$
```

Listing 1: Pseudo code for PFS

In Listing 1, the pseudo code is given for the node implementation of PFS. In this algorithm, node A is the sender of a flooding message m, while node B is a receiver. $N(x)$ denotes the set of neighbors of node x. $C(m)$ is the uncovered neighbor set of message m on node B, i.e., the list of neighbors of node B not yet covered by a transmission of the message m. $d(m)$ is the amount of time the node needs to wait before eventually retransmitting the packet, which is calculated by the $\mathrm{RAD}_{\mathrm{PFS}}$ function based on the neighbor number of the sender, the number of uncovered neighbors, and the flooding packet length. If the received message is not a new message, the node removes the covered neighbors from its uncovered neighbor set for message m. When the delay expires, the node checks if the set is empty. If so, it drops the message. Otherwise, it retransmits the message.

5.3.2 The design of RAD

So far, PFS is nothing more than a combination of FSP and SBA. The main novelty of PFS is in the design of the RAD and how the nodes schedule their retransmissions. To achieve good performance, it is important that nodes, which have many uncovered neighbors (neighbors that have not yet received the flooding message), retransmit first. One good option is to let nodes count how many uncovered neighbors they have (i.e., $|C(m)|$) after receiving a message for the first time and schedule their retransmissions ordered by that result. The one with most uncovered neighbors retransmits first. In this way, PFS tries to let nodes with more uncovered neighbors retransmit first, which is different from SBA. In SBA, a node with many neighbors is only more likely to retransmit first and does not take into account whether its neighbors already are covered or not.

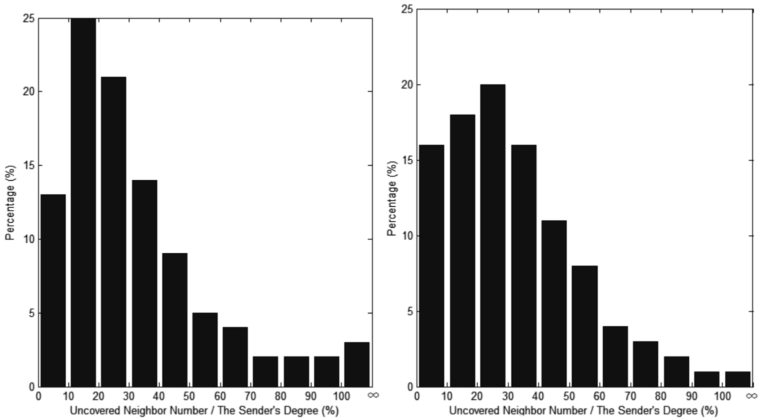
PFS assigns a fixed time interval, which is equal for all receiving nodes. The interval is divided into slots of equal lengths, where receiving nodes with many uncovered neighbors schedule their retransmissions in the earlier slots and nodes with few uncovered neighbors schedule their retransmissions in the later slots. To properly determine the RAD and the slot assignment process, we need to answer a few questions.

The first question is how long the total time interval should be. From [225], it can be concluded that the number of retransmissions generated by one transmission is rarely more than 5 if the self-pruning works as it should[1]. This means that there is no reason for the RAD to be longer than 5 times the transmission time of the flooding message plus some extra time to accommodate for other concurrent transmissions. In sparse networks, when the estimation is very low, even 5 transmissions are too much. In those cases, we make the RAD shorter.

The next question is how the nodes decide how many slots there should be and in which slot to retransmit. Due to the use of one-hop hello messages, the nodes have very limited knowledge about their neighbors. For instance, no node knows the other nodes' number of uncovered neighbors and can not know if it has the most. Therefore, a good estimation is needed. This estimation, which we call *estimated maximum number of uncovered neighbors* or EMNUN, could for instance be the number of neighbors of the sender, which is given in the flooding message. However, the largest additional coverage of a retransmission is at most 61 % of the whole transmission coverage [225] if we assume a circular coverage area. Hence, a better estimation is 60 % of the sender's number of neighbors. For example, if the sender has 23 neighbors, the estimation will be $\lceil 0.6 \cdot 23 \rceil = 14$ uncovered neighbors.

To verify the suitability of this estimation, we simulated PFS in ns-2 [159]. More details about the simulation is given in [99]. Figure 5.1 shows the simulation result for the number of uncovered neighbors that a node has, after receiving a flooding message for the first time in a random network with two different densities. The x-axis is the number of uncovered neighbors divided by the sender's neighbor number expressed in percent. The y-axis is the distribution of receiving nodes. From this figure, we can see that less than 13% of the nodes have more uncovered nodes than 60% of the sender's neighbor number in both cases. This indicates that the 60% estimation is a reasonable estimation. By this calculation, each receiving node can estimate the maximum number of uncovered neighbors of all receiving nodes. Since they all use the same data, they will arrive at the same estimation. Nodes

[1] At least in 2-dimensional networks



(a) Average Node Degree 9.1

(b) Average Node Degree 16.1

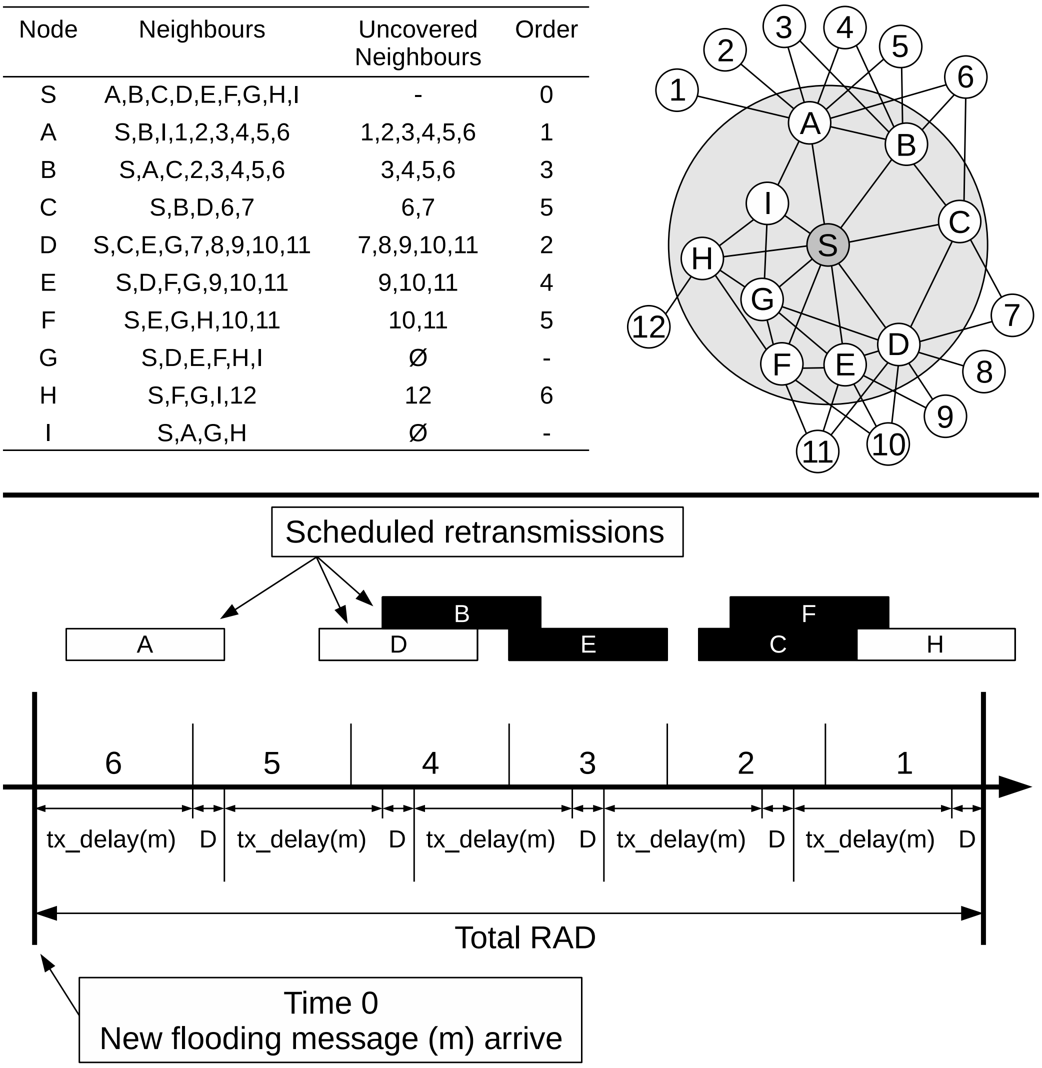
Figure 5.1: Number of Uncovered Neighbors Distribution

with this amount of uncovered neighbors or more transmit in the first slot, nodes with one less uncovered neighbor transmit in the second slot and so on. Nodes with only one uncovered neighbor transmit in the last slot (unless they self-prune).

Figure 5.2 shows an example network where a source node (S) initiates a flooding message. The EMNUN is estimated to be 6 due to the 9 neighbors of S. Upon reception, two nodes (G and I) are able to self-prune immediately. The remaining 7 nodes have uncovered neighbors and may assign their retransmissions as shown in the lower part of the figure. Note how nodes with more uncovered neighbors schedule themselves earlier in the RAD. When nodes schedule to retransmit early (e.g., A) start to retransmit, later nodes will be able to self-prune (indicated by black in Figure 5.2). As a consequence, enough room is created in the RAD for all retransmissions. In the end, the retransmissions of A, D, and H are enough to cover the whole network.

However, it is important to notice that too many slots results in extremely small slots and this can sometimes be a problem. Many nodes may then schedule their retransmissions in the same slot. Transmissions may not collide (if the MAC does it job), but one transmission may not cancel the other retransmissions due to the short time between transmissions. This is mainly due to the MAC protocol deferring transmission due to a busy channel or the time needed between the reception of a message and the processing of it. In the mean time, the decision to retransmit is already taken. Therefore, we introduce an upper limit to the number of slots, which we denote as MAX_SLOTS. Later, we will show that increasing MAX_SLOTS beyond a certain point does not make the self-pruning any more efficient.

If EMNUN is less than or equal to MAX_SLOTS, a node with the same or higher number of uncovered neighbors will retransmit in the first slot, nodes with one less in the second slot, and so on. On the other hand, if EMNUN is more than MAX_SLOTS, the slot that best matches the node's number of



Node	Neighbours	Uncovered Neighbours	Order
S	A,B,C,D,E,F,G,H,I	-	0
A	S,B,I,1,2,3,4,5,6	1,2,3,4,5,6	1
B	S,A,C,2,3,4,5,6	3,4,5,6	3
C	S,B,D,6,7	6,7	5
D	S,C,E,G,7,8,9,10,11	7,8,9,10,11	2
E	S,D,F,G,9,10,11	9,10,11	4
F	S,E,G,H,10,11	10,11	5
G	S,D,E,F,H,I	Ø	-
H	S,F,G,I,12	12	6
I	S,A,G,H	Ø	-

Figure 5.2: Example RAD with slots and slot assignments

uncovered neighbors is selected. The retransmission is scheduled uniformly inside the slot. The details are given in the pseudo code of the RAD_{PFS} function in Listing 2.

In the pseudo code, num_slots is the number of slots, slot is the slot number (0 to num_slots −1) selected by the node, and slot_len is the slot length in seconds. Both num_slots and slot_len are the same among all nodes receiving from the same sender. Uniform$(0, 1)$ returns a random value uniformly distributed between 0 and 1. tx_delay$(|m|)$ denotes the transmission time of message m and D is the extra delay introduced to handle concurrent traffic. The first part of the code determines the number of slots and which slot to select. The second part calculates the length of the time interval and thereby the length of the slots themselves. Here we also use 60 % of the sender's number of neighbors to guess whether we can anticipate less than

5 retransmissions. If so, we shorten the time interval by not enlarging the slot_len beyond tx_delay($|m|$) + D.

```
If $|N(A)| = 0$
    $\text{num\_slots} = 1$
    $\text{slot} = 0$
Else If $0.6 \cdot |N(A)| < \text{MAX\_SLOTS}$
    $\text{num\_slots} = \lceil 0.6 \cdot |N(A)| \rceil$
    $\text{slot} = \text{num\_slots} - |C(m)|$
Else
    $\text{num\_slots} = \text{MAX\_SLOTS}$
    $\text{slot} = \text{MAX\_SLOTS} - \left\lceil \frac{\text{MAX\_SLOTS} \cdot |C(m)|}{0.6 \cdot |N(A)|} \right\rceil$
If $\text{slot} < 0$
    $\text{slot} = 0$

If $\lceil 0.6 \cdot |N(A)| \rceil > 5$
    $slot\_len = \frac{5 \cdot (\text{tx\_delay}(|m|) + D)}{\lceil 0.6 \cdot |N(A)| \rceil}$
Else
    $\text{slot\_len} = \text{tx\_delay}(|m|) + D$
Return $(\text{slot} + \text{Uniform}(0,1)) \cdot \text{slot\_len}$
```

Listing 2: Pseudo code of the RAD_{PFS} function

The implementation of the PFS algorithm in Listing 1 and Listing 2 was quite simple and straight forward. The resulting code is not computationally heavy. All code was written in fixed point arithmetics and could easily run on our experimental platform, which only has a 8 MHz 16-bit microcontroller. Hence, we believe that the PFS implementation can be implemented and run on all relevant hardware.

5.3.3 Early retransmissions

A limit of MAX_SLOTS number of slots does not always work as one would wish. There are cases when many nodes will select the same slot and this would require the slot length to be very big. Examples are networks with inhomogeneous density or a lot of border effect (nodes on the border of the network have fewer neighbors than nodes in the center of the network). Another example is when only the initiator of a flooding message failed to detect a neighbor (e.g., hello message was lost) in a fully connected network causing everybody else in the network to believe that the sender covered all but that node. Since all other nodes in the network can cover that last node, everyone schedules a retransmission in the last slot.

This problem cannot be fully solved by MAX_SLOTS or extending the slot length by increasing D since this will adversely affect the normal operation of PFS. A better approach is to randomly select one or a few of the nodes to retransmit earlier than their number of uncovered neighbors indicate. This will frequently allow all the remaining nodes to self-prune. However, the nodes need to independently decide to retransmit early without conferring with the other nodes. This can be done by letting nodes scheduled for transmission in one of the last slots to transmit early with a probability p.

The selection of p is all but trivial. Too low a p may cause none of the nodes to retransmit early and too high a p will cause too many to retransmit unnecessarily. The parameter p also depends on the situation, such as the number of nodes that schedule a retransmission in the same slot.

To find the optimal p, we assume n contesting nodes. Without any early retransmissions and assuming that one retransmission makes the others self-prune, the expected number of retransmissions is:

$$E[\#\,\text{Retransmissions}(\text{no}\,\text{modification})] = 1 + \frac{n-1}{slot\_len/STT(m)}$$

where STT(m) is the expected single trip transmission time for message m. During that time, there may be other retransmissions. Those retransmissions will take place since the nodes are not able to self-prune in time. Only scheduled retransmissions later than STT(m) are able to self-prune. Hence, we may see many unnecessary retransmissions if n is large.

If we implement an early retransmission function, the expected number of retransmissions becomes different. Assuming that an early retransmission cancels all the non-early retransmitters but early retransmissions never cancel each other, we get:

$$\begin{aligned} &E[\#\,\text{Retransmissions}(\text{with}\,\text{early}\,\text{retransmission})] = \\ &\qquad P(\geq 1\,Early) \cdot (1 + E[\#Early(n-1)]) + \\ &\qquad P(=0\,Early) \cdot E[\#Retransmissions(no\,modification)] \end{aligned} \tag{5.1}$$

where:

$$\begin{aligned} &P(=0\ \text{Early}) = (1-p)^n \\ &P(\geq 1\ \text{Early}) = 1 - P(=0\ \text{Early}) = 1 - (1-p)^n \\ &E[\#\text{Early}(n-1)] = p \cdot (n-1) \end{aligned}$$

With this formula, we can calculate the expected number of retransmissions for different scenarios and with different parameter p. Figure 5.3 shows the results for some different p when assuming a slot_len / STT(m) ratio of 5. None of these curves gives an optimal result at all times. Each n has its own most optimal p. While the optimal p clearly depends on n, we found that the impact of the slot_len to STT(m) ratio is less pronounced. Hence, we should at least make p dependent on n.

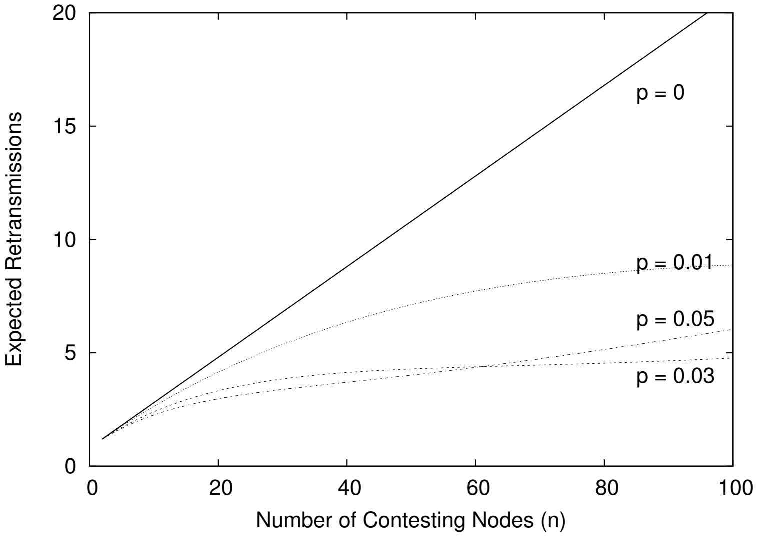


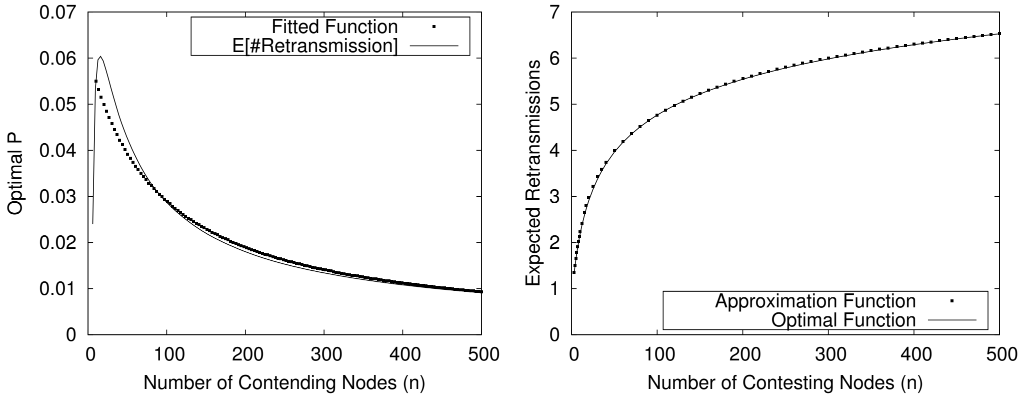
Figure 5.3: Expected retransmissions and p

To go from here to a solution that can find the optimal p, we first need to estimate the number of contesting nodes (n) and then identify an easy way to calculate p using that estimate. For the n estimate, we propose to use the EMNUN again, which probably is the best we can do with the limited information each node has. To define the function that finds the optimal p based on formula (5.1) does not lead to a simple enough function. Since accuracy of this function is not of utmost importance, we instead propose to use a simple approximation function. This was achieved by numerically calculating the optimal p for every $n > 10$. When n is small ($n < 10$), there is very little use of this mechanism anyway.

We propose $p = \text{EARLY\_P}\,/(n + 90)$, which is extremely simple and only requires two arithmetic operations. EARLY_P is a constant that can be tuned. 5.5 seems to be a very good fit according to our analysis as shown in Figure 5.4(a), but we will also simulate and measure what is best in more realistic situations, since some of the assumptions are too simple. For instance, a single early retransmission may not cancel all scheduled retransmissions and the expected number of retransmitting neighbors is likely to be different than the number of contesting nodes n.

As shown in Figure 5.4(a), the proposed function does not accurately estimate the optimal value of p when n is small. However, this has very little impact on the retransmissions and that is what really matters. Figure 5.4(b) shows how the expected number of retransmissions will look like according to formula (5.1) for the proposed function and when using the most optimal values. It is clear that the estimation is very good. Similar results are achieved also when the slot_len / STT(m) ratio is varied around 5.

To implement this enhancement in PFS, the code snippet in Listing 3



(a) Optimal p and approximation function

(b) Expected retransmissions using approximation function

Figure 5.4: PFS hello protocol measurement results

should be inserted right before $d(m)$ is calculated by the RAD_{PFS} function in Listing 2.

It only works when there may be more than 10 contesting nodes and the node itself has scheduled a retransmission in the second half of the RAD. If it decides to retransmit early, the slot will be changed to the corresponding slot in the first half of the RAD. That is, given 5 slots, a node changes from slot 5 to 2 or from 4 to 1, if it retransmits early.

```
If $|N(A) > 10$ and slot $> \lfloor$num_slots$/2\rfloor$
  If Uniform$(0,1) < \dfrac{\text{EARLY\_P}}{|N(A)|+90}$
    slot = slot $- \lceil$num_slots$/2\rceil$
```

Listing 3: Pseudo code of the early retransmission function

To better understand how the early retransmission functionality works, we simulated it in ns-2 [159] using 50 nodes. Each node was placed randomly in a rectangular (ratio 4:9) flat area. In each experiment, 200 flooding messages were generated from random nodes. Each simulation was repeated 20 times with the 99 % confidence interval shown. More details about the simulation environment is given in Section 5.6.

Figure 5.5 shows the results of several hundreds of simulations using different network densities and different values of EARLY_P. Each curve represents a certain network density measured by the average node degree. It is clear that a EARLY_P around 12.5 can improve the performance when the network is dense. As the network becomes sparser, the benefit diminishes and may even degrade the performance. However, the degradation in sparser networks is much less than the improvement in denser networks.

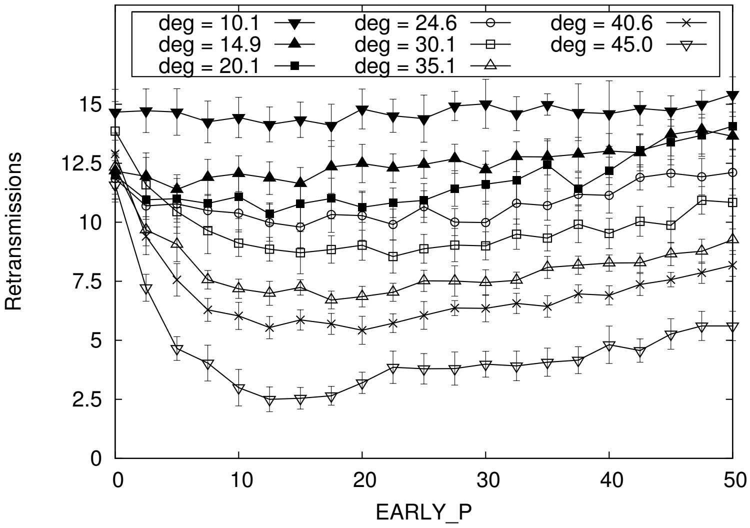


Figure 5.5: PFS simulation of EARLY_P

5.4 The Test Bed Environment

Most of the optimized flooding research has been based on simulations. However, it is also important to verify the performance in real networks due to the complex characteristics of wireless networks that cannot be captured in a simulator. In fact, the inaccuracy of wireless networks simulations has been known for some time now [29]. Multi-path fading, noise, interference, and hidden terminals are examples of factors that are hard to accurately reproduce in a simulator. Consequently, most simulations use a perfect circular transmission area. However, links are frequently somewhere in between good and bad or experience strong asymmetric behavior [45]. Unfortunately, many flooding protocols are designed without considering these characteristics. They perform very well in the simulator environment, but the question remains; how do they perform under realistic circumstances?

These are all reasons to evaluate PFS in more realistic circumstances, such as in a real wireless multi-hop network. To properly test the performance of PFS, we compared it to Blind Flooding and CBB. In Section 5.6, we also compare these test bed measurements with the results obtained in a simulator. However, first we introduce the test bed that we used to compare these protocols.

5.4.1 The hardware and software platform

For our validation, we used t-mote Sky (based on Telos Revision B) [152] from Moteiv Corporation, which is a sensor mote platform for extremely low power, high data-rate, wireless sensor network applications. The reasons to use sensor motes were cost efficiency and ease of large scale deployment compared

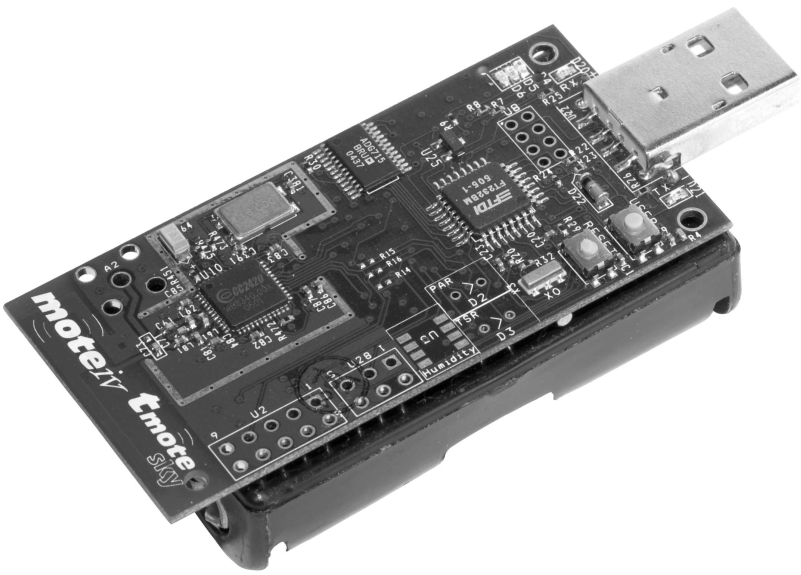


Figure 5.6: The t-mote Sky sensor mote

to if we would have used laptops or PDAs equipped with WLAN. The wireless technology used by t-mote Sky is 2.4 GHz IEEE 802.15.4 [87], which is a wideband technology not very different from most wireless technologies of today, including IEEE 802.11. 2.4 GHz IEEE 802.15.4 uses direct sequence spread spectrum (DSSS) RF modulation with a data rate of 250 kbps.

A photo of the t-mote sky sensor mote is shown in Figure 5.6 while its specification is given in detail in [152]. In short, it uses a CC2420 Chipcon Wireless Transceiver [221], which is connected to an integrated onboard antenna. The microcontroller is an 8 MHz Texas Instruments MSP430 that can run TinyOS [222] and is equipped with 10 kB RAM and 48 kB flash memory. Further, the t-mote has a USB interface that can be used to upload program code, experiment configuration parameters, and to download collected experiment data.

The IEEE 802.15.4 frame format begins with a 5 bytes synchronization header (SHR), followed by 7 bits (one bit is reserved) defining the number of bytes in the MAC Protocol Data Unit (MPDU). In our experiments, the MPDU consisted of a 9 bytes header that contained frame type, packet sequence number, addressing information, etc. At the end of the packet, there is a 2 bytes CRC field. All in all, this means that a packet consisted of a payload plus 17 bytes of headers.

We used the default MAC protocol provided by Boomerang 2.0.4. Boomerang is a software package developed by Moteiv that is based on TinyOS and tailored for the t-mote platform. The default MAC in Boomerang instructs the radio to continuously listen for transmissions. When transmitting, a mote uses the clear channel assessment (CCA) function defined in [182] to determine whether the channel is idle or not. If there is an ongoing trans-

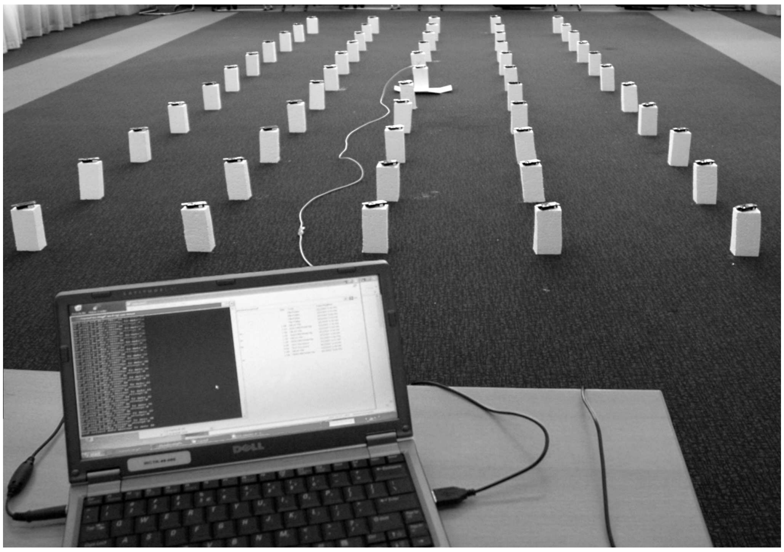


Figure 5.7: Photo from the experiment setup

mission, the mote waits a random time (uniform between 8 and 1024 μs) and tries again. After 8 unsuccessful attempts, it gives up and drops the packet.

5.4.2 Experiment setup

To avoid interference of external systems, we monitored the spectrum and chose a non-occupied channel. Then, we placed 50 motes in a matrix topology with 10 motes lined up in 5 rows. The distance between neighboring motes were the same, but varied between 0.3 m and 2.0 m for the different experiments. Each node was elevated about 20 cm above the floor using blocks of polystyrene foam in order to avoid the worst kind of multi-path interference. To avoid the need of a very large hall, we reduced the transmission power to almost minimum (-24 dBm). A photo of an example scenario is shown in Figure 5.7. One of the motes in the center was connected to a laptop with a USB cable. The laptop was used to control the experiment by instructing the USB-connected mote to transmit instructions to the other motes using full power. With full power from a centrally located mote, all motes could be reached in one hop. The same technique was also used to collect the measurement data.

Since a larger distance between neighboring motes meant a sparser network, we defined a measure for the network density. It is basically the average node degree (number of direct neighbors), but also considers the packet delivery ratio of the links. If a particular link experienced a packet delivery ratio of 40 %, we counted that as 0.4 links. We did this because a flooding protocol should be able to exploit the fact that packets can be sent on this link with a probability of 40 %. For unicast protocols, such links would probably be useless and should be avoided, but for flooding, they are still useful.

Therefore, we used the following definition as a measurement of the network density:

$$Avg.\,Node\,Degree = \frac{1}{N}\sum_{x=1}^{N}\sum_{y=1}^{N} R_{x,y} \tag{5.2}$$

where N is the number motes, $R_{x,y}$ is the packet delivery ratio from mote x to mote y and $R_{x,x} = 0$. A higher value means a denser network with 49 as the maximum in our 50 motes networks. For each network scenario, we measured the average node degree by letting each mote transmit 500 packets. At the same time, each mote listened for packets from other nodes so that the packet delivery ratio for all node pairs could be estimated. The CCA functionality in combination with a sufficiently long packet generation interval was enough to practically eliminate collisions.

Each measurement started with the laptop instructing the USB-connected mote to send a configuration and start of experiment packet with full power to all the nodes in the network. Upon receiving this message, every node lowered their transmission power to the configured level and started to exchange some few hello messages. Depending on the configuration, each node generated one or three hello messages with a hello interval uniformly distributed between 0.9 s and 1.0 s. These hello messages only contained the mote's address and were used to detect one-hop neighbors. For PFS, the hello protocol is a crucial part that may affect its performance. Therefore, we tested three different approaches to determine a mote's neighbors:

1. A lenient method, where it is sufficient to receive one out of the last three hello messages to accept a mote as a neighbor. Hence, some included neighbors are difficult to reach.

2. A standard method, where the reception (or non-reception) of the last hello message is used to determine if a mote is considered a neighbor.

3. A strict method, where all the last three hello messages must be received to accept a neighbor. This method creates the smallest neighbor set.

After the hello messages, a small waiting time of 0.5 s followed in order to avoid collisions between hello and flooding messages. Then, one mote, randomly selected, initiated a flooding. Every node in the network responded to this according to the chosen flooding protocol and configuration. Whenever a mote received the flooding message for the first time or retransmitted the flooding message, the time was stored in the mote's memory.

When the flooding finished, we collected the data. On behalf of the laptop, the USB-connected mote queried each mote for their measurement data, such as when and if it received the flooding message and when and if it retransmitted it. All this data was compiled at the laptop to calculate the

Table 5.1: Network scenario connectivities

Network scenario	Avg. node degree
Fully connected	48.1
Very Dense (0.3 m)	44.4
Dense (0.6 m)	32.3 - 37.5[2]
1.0 m	19.3
1.5 m	11.0
Sparse (2.0 m)	6.0

reachability, retransmissions, and delay of this single flooding. The measurement was then repeated until a sufficient number of floodings had been conducted to make statistically significant conclusions.

5.5 Experimental Results

To properly compare PFS with CBB and Blind Flooding, the experiment was conducted in two phases. The target of the first phase was to determine the optimal parameters for each of the tested flooding protocols, whereas in the second phase, we compared the three flooding protocols using those parameters in six network scenarios with different density.

Each measurement was repeated 50 times. In all the resulting graphs, we show both the average value and the 90 % confidence interval. For the network scenarios that we used in our experiments, the average node degrees according to formula (5.2) are shown in Table 5.1. The experiments were conducted at different times and that meant that the connectivity sometimes varied a bit even though we tried hard to maintain the same environment. The fully connected scenario was done with full transmission power, since it was impossible to achieve full connectivity otherwise. However, all graphs and comparisons between configurations and protocols were done at the same time using the same network scenario and hence with minimal scenario-induced differences.

To compare the flooding protocols, we use the following three measurements in the remaining of this chapter:

Reachability This measurement evaluates a protocol's reliability. It is represented by the delivery ratio of a flooding message. For example, if there are 50 nodes in a network and a node floods a message using a certain flooding protocol resulting in 42 nodes receiving this message, then we say that the reachability is 42/49 = 85.71%.

[2] 37.5 was measured in phase one in a large meeting room, while 32.3 was in phase two in a sports hall

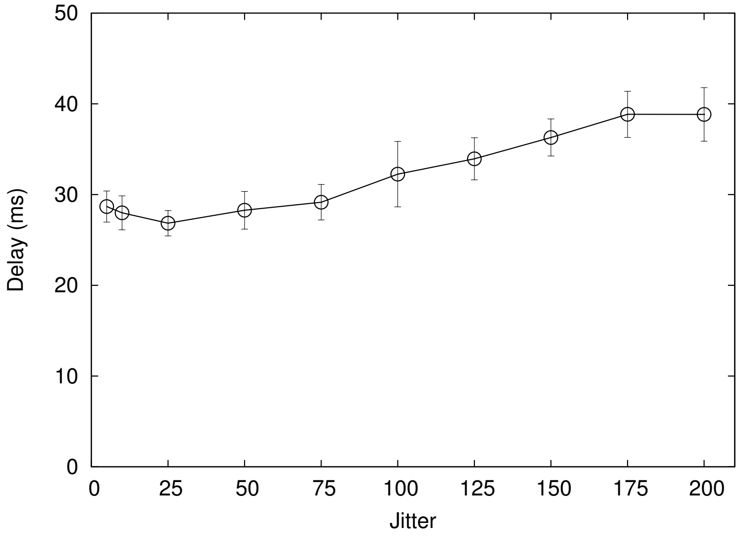


Figure 5.8: Blind Flooding

Retransmissions This is a measurement that measures the number of retransmitting nodes for flooding a message. Other messages, such as hello messages are ignored in this measurement. It measures the efficiency of the protocol.

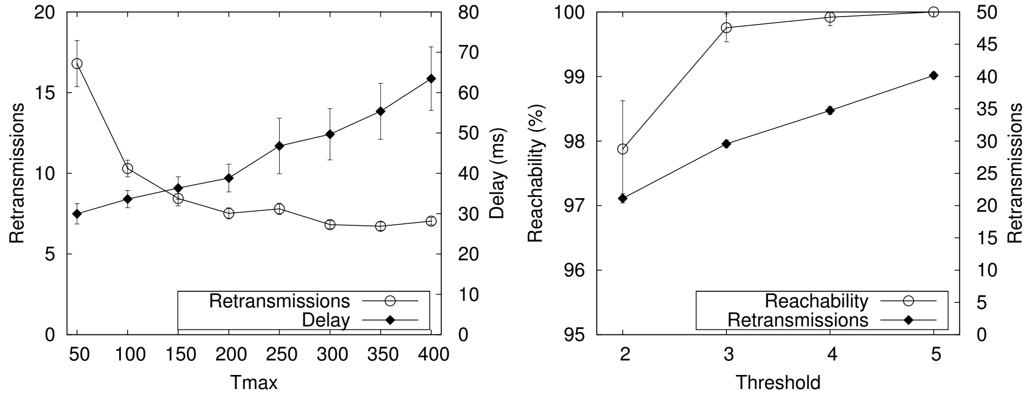
Delay The delay counts the time interval from the moment that the source node sends a flooding message until the moment that the last node in the network receives this message. Because not all messages reach all nodes, the last node means the last one which received the message.

5.5.1 Optimizing the protocol parameters (phase one)

In this section, we present the results of phase one, where we tried to determine the most optimal parameters for each of the protocols. We present the results protocol by protocol, staring with Blind Flooding.

Blind Flooding

We started with determining the jitter parameter of Blind Flooding. This was done in the dense network scenario. We varied the jitter parameter from 200 ms down to 5 ms, while measuring reachability, retransmissions, and delay. Reachability and retransmissions were always 100 %, while the delay decreased down to its lowest point at a jitter of 25 ms after which it slightly increased as shown in Figure 5.8. Therefore, we opted for a maximum jitter of 25 ms.



(a) RAD measurements

(b) Threshold measurements

Figure 5.9: CBB measurement results

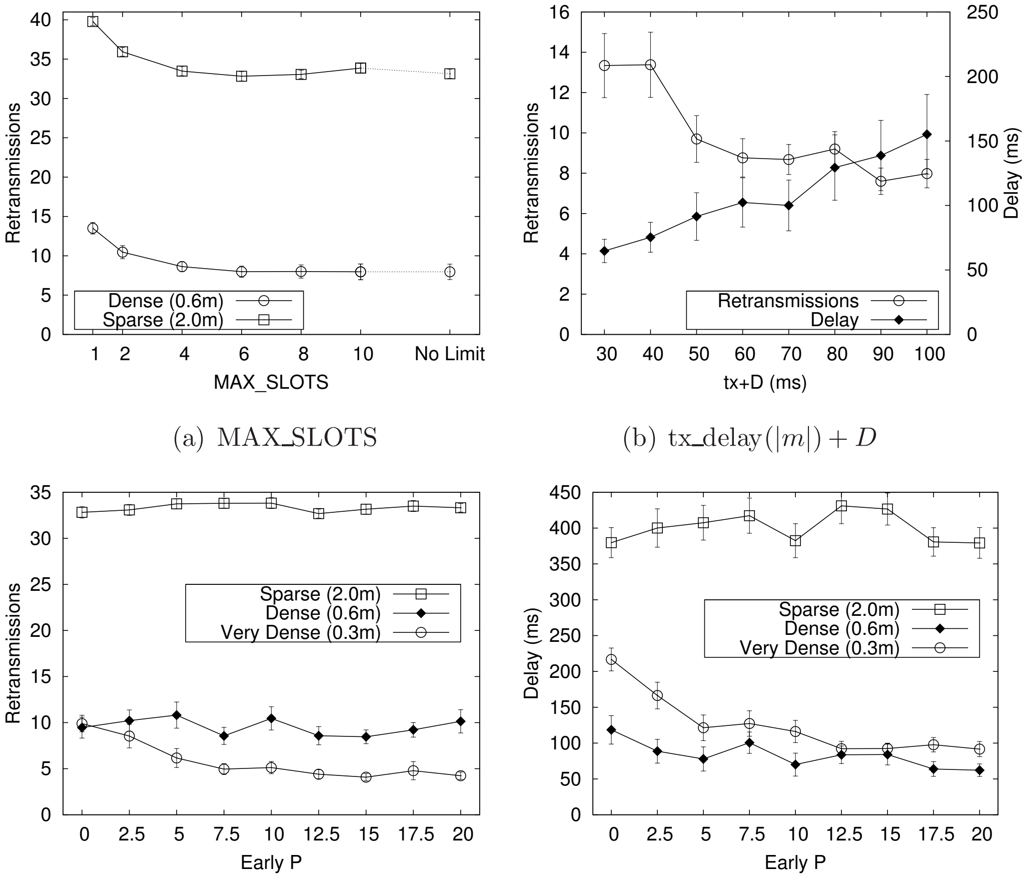
CBB

Next, we continued with CBB. Two parameters needed to be determined; the maximum RAD length (T_{max}) and the threshold. To find the optimum T_{max}, we used a threshold of 3, which is a value known to work well from simulations. We tried different values of RAD in a dense scenario and measured the number of retransmissions and delay. The reachability was always extremely good in this dense scenario. As T_{max} became smaller, the retransmissions increased as demonstrated in Figure 5.9(a). This is due to the delay between the retransmission decision and the actual retransmission. This delay is quite large in the t-motes and hence causes the retransmissions to increase rapidly when T_{max} becomes too small and, in dense networks, this issue becomes more severe. On the other hand, with a larger RAD, the delay also becomes larger. From Figure 5.9(a), we can see that when the RAD is 200 ms, there is very little improvement in the retransmissions if further increased, while the delay continues to grow. Hence, we select $T_{max} = 200$ ms.

Then we studied the CBB threshold in the sparse network scenario. The sparse scenario was chosen, because it is the least connected and the most difficult to reach all nodes. We measured the reachability and the retransmissions, which are shown in Figure 5.9(b). As can be seen, a threshold of 2 causes the reachability to suffer, while a value of more than 3 does not improve much. Instead, it only increases the number of retransmissions. Hence, we chose 3 as the best CBB threshold.

PFS

We needed to determine MAX_SLOTS, tx_delay($|m|$) + D, EARLY_P, and what is the best hello protocol for PFS. First, we tested PFS with different MAX_SLOTS in both the dense and the sparse network scenarios. In the



(a) MAX_SLOTS

(b) tx_delay$(|m|) + D$

(c) Retransmissions and EARLY_P

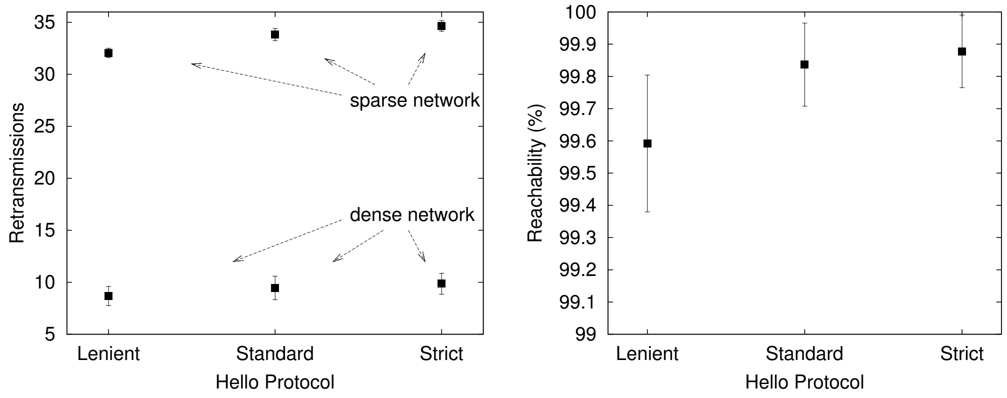
(d) Delay and EARLY_P

Figure 5.10: PFS measurement results

sparse network scenario, we used a tx_delay$(|m|) + D$ of 60 ms which was known to be a good trade off between delay and performance from preliminary experiments (this is further verified in Figure 5.10(b)). However, in the dense scenario we used a very large tx_delay$(|m|) + D$ (almost 200 ms) to avoid the effect of too crowded slots, which can happen in denser networks. Furthermore, we used the standard hello protocol and switched off the early retransmission feature. The results are shown in Figure 5.10(a).

When MAX_SLOTS $= 1$, the priorities among the retransmitting nodes are in practice not used. That is, the order is completely random. However, already after using two slots, we can see an immediate improvement. The improvement continues until around 6, where it levels off or even degrades in the sparse case. The reachability (not shown) remained unaffected and constantly over 99.59 %. Hence, 6 seems like the best option since we also do not want to make the slots too small as explained earlier.

Then we continued with tx_delay$(|m|) + D$. Since, all flooding messages were the same, the transmission time tx_delay$(|m|)$ was constant. Therefore,



(a) Retransmissions (b) Reachability in sparse network

Figure 5.11: PFS hello protocol measurement results

we could just as well determine the entire sum of tx_delay($|m|$) + D. This was done in the dense scenario because of the same reasons as for CBB earlier. We measured retransmissions and delay and the results are shown in Figure 5.10(b). We used the same parameters as in the previous experiment, but with MAX_SLOTS = 6. We can see that the reduction of retransmissions starts to level off after a tx_delay($|m|$) + D of about 60 ms, while the delay continues to increase. The reachability (not shown) was as near to perfect as one can come. Because of this, we chose 60 ms.

The next parameter to be studied was EARLY_P and the improvements brought by the early retransmission feature. We conducted experiments with EARLY_P varied between 0 (off) and 20 in three different network scenarios, namely the sparse, the dense, and the very dense network scenarios. The benefit of the early retransmission feature should be most pronounced in a very dense network, which can be seen from the results in Figure 5.10(c). In sparser networks, this feature is not useful and is also less used. For the dense and the sparse network scenarios, no trend can be seen which indicates that the early retransmission feature did neither improve nor adversely affect the performance. Anyhow, we believe that the EARLY_P should be set as small as possible and since there is almost no gain from about 10.0 and onwards in any of the network scenarios, that seems a good value. In Figure 5.10(d), the delay in the same experiment is shown. Here it is clear that the delay also decreases when the early retransmission feature is being used. The reachability (not shown) again remains unaffected and constantly high.

Finally, we investigated whether the lenient, standard, or strict hello protocol is best for the performance of PFS. This was done in both the dense and the sparse scenarios. The number of retransmissions of PFS in both scenarios are shown in Figure 5.11(a), while the reachability in the sparse scenario is shown in Figure 5.11(b). The reachability was 100 % in the dense scenario (not shown) for all hello protocols. In the sparse scenario, a slight, but not statistically significant, decrease could be noticed for the lenient (average 99.6 %) and the standard method (average 99.8 %) compared to the strict method (average 99.9 %). For the delay (not shown), the strict method indicated a smaller average delay in both the scenarios compared to the lenient and the standard methods. The latter two had almost the same average delay, however, not statistically significant. In the end, we felt the lenient method is able to perform the best since it reduced the number of retransmissions while maintaining very high reachability.

5.5.2 Flooding protocol comparison (phase two)

In phase two, we compared the three protocols with each other using the parameters deduced from the previous experiments, which are summarized in Table 5.2. The values chosen might still not be optimal despite all our efforts. Sometimes it is quite hard to make a good choice when there is a trade off between two performance measurements. Others may have different requirements and preferences. Furthermore, one value works best in one scenario, while another works best in another scenario. However, a flooding protocol should work well in all scenarios with one set of parameters and not require to be tuned to the given situation. Hence, one fixed set of parameters was chosen that should perform well in all scenarios if possible.

We tested the protocols in six different network scenarios starting from a fully connected network and then gradually making it sparser. All experiments in phase two were done in a large sports hall. The results of the retransmissions, reachability, and the delay measurements are shown in Figure 5.12, Figure 5.13, and Figure 5.14 respectively. On the x-axis in all three graphs, we used the average node degree corresponding to the measured scenario going from dense on the left to sparse on the right.

In Figure 5.12, we show the retransmissions generated by each of the protocols in the various scenarios. The retransmissions of CBB and PFS follow each other quite well and are both significantly better than Blind Flooding also in real wireless multi-hop networks. While PFS has an advantage when the network is fully connected, CBB performs slightly better in the medium density scenarios. This demonstrates the difficulty in making self-pruning perform well. Despite all our efforts to optimize the performance of PFS with a carefully designed RAD, CBB still performs better. We do believe that finding even more efficient RAD designs will prove to be very difficult. Instead of continuing down that path, we decided to see what happens when we combine PFS with CBB. This new protocol, which we call Counter-Based PFS (CB-PFS), is also shown in the remaining figures. It simply works exactly as PFS, but with a counter like CBB. CB-PFS can refrain from retransmitting either due to self-pruning according to PFS or due to the counter exceeding the threshold according to CBB. The parameters used for CB-PFS were exactly the same as for PFS and CBB. As expected, CB-PFS achieves

Table 5.2: Parameters used in comparison

Protocol	Parameter	Value
All	Number of nodes	50
All	Total flooding packet size	42 bytes[3]
All	Total hello packet size	19 bytes
All	Number of floodings per scenario	50
BF	Jitter	25 ms
CBB	T_{max}	200 ms
CBB	Threshold	3
PFS	MAX_SLOTS	6
PFS	tx_delay($\|m\|$) + D	60 ms
PFS	EARLY_P	10.0
PFS	Hello protocol	Lenient
PFS	Hello packet interval	0.9 - 1.0 s

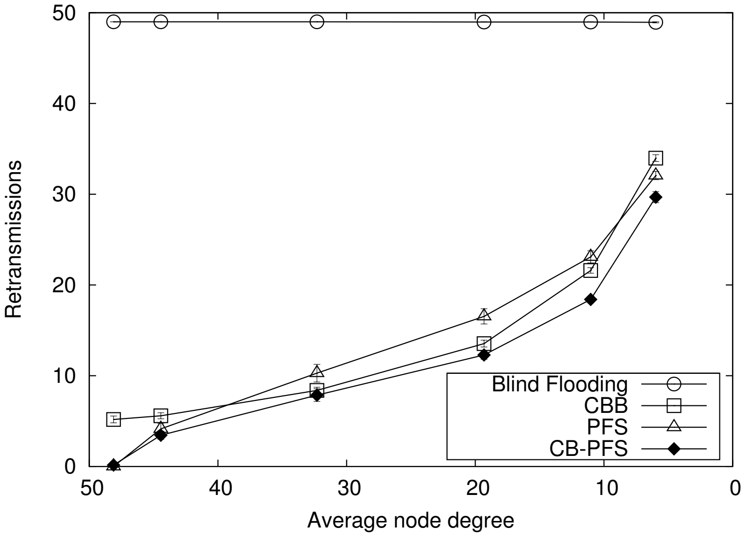


Figure 5.12: Retransmissions

fewer retransmissions than both PFS and CBB in all scenarios.

Figure 5.13 shows the reachability of all four tested flooding protocols. All protocols struggle to reach all nodes when the network becomes too sparse. Even Blind Flooding struggles to achieve full reachability. However, we must

[3] The payload was 23 bytes for BF/CBB, but only 15 bytes for PFS due to extra overhead

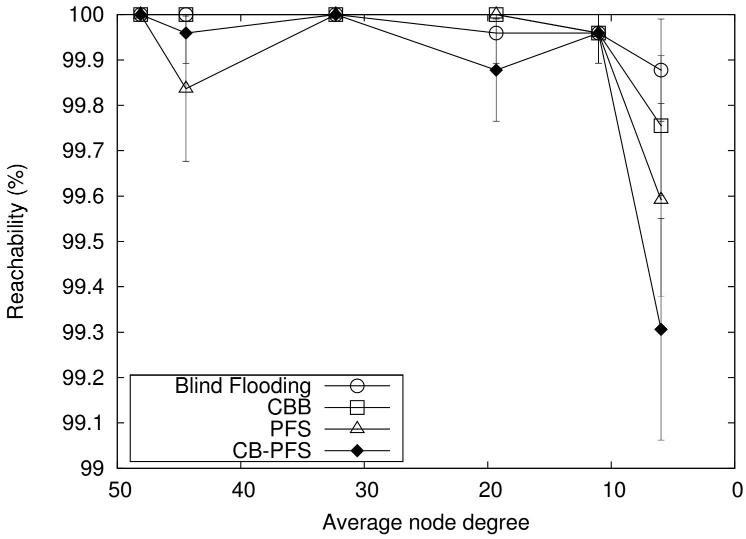


Figure 5.13: Reachability

note that the reachability is extremely high in all cases. A reachability of 99.96 % means that only one node failed to receive only one of the 50 flooding messages. Even CB-PFS, which is the most efficient protocol, achieves a reachability above 99.3 %.

The last measurement is delay and is presented in Figure 5.14. Here, Blind Flooding is clearly better. We probably could have improved the delay performance for PFS and CB-PFS by reducing the D parameter. However, that would, at the same time, have a negative effect on retransmissions. The cause for this is the relatively long time it takes the t-motes to process received packets and cancel a scheduled retransmission. If we could improve this for the t-motes, we would be able to significantly improve the delay performance for both PFS and CB-PFS as well as for CBB.

5.5.3 Measurement conclusions

Blind Flooding is the protocol with the highest reachability and the shortest delay in our experiments. However, the cost for this is the large amount of unnecessary retransmissions. On the other hand, CB-PFS reduces the amount retransmissions to such a low level that the reachability starts to drop. Fortunately, CB-PFS still manages to reach the nodes in almost all of the cases.

Despite its simplicity, CBB performs exceptionally well. PFS, on the other hand, requires quite some complexity to achieve a good performance. This is not to say that the calculations done in PFS are heavy. The calculations are actually very simple and posed no problems for our sensor motes to compute and hence should rarely be a problem. Instead, the complexity lies in the number of parameters and the various mechanisms working in PFS

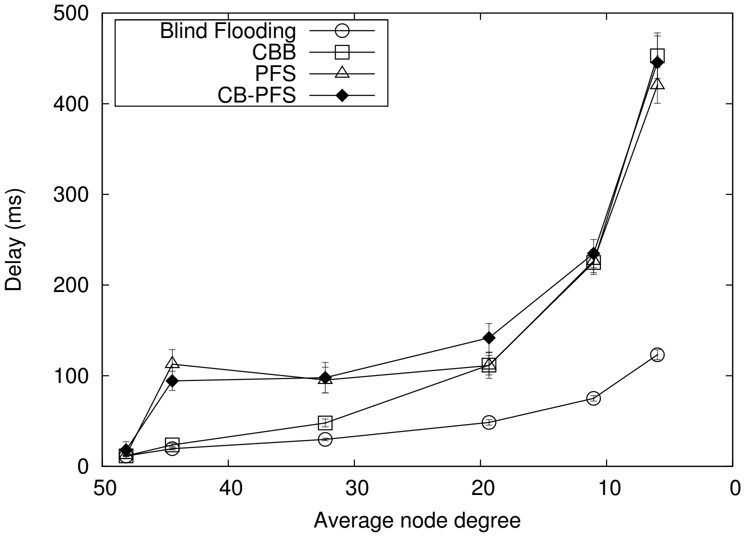


Figure 5.14: Delay

and this makes it difficult to find the most optimal setting. We believe that it is still possible to identify better parameters.

For sparse networks and networks with few nodes, PFS or CB-PFS perform better than Blind Flooding and CBB. This makes PFS or CB-PFS very suitable for use in Clusters, since Clusters are expected to be fully connected if using one single wireless communication technology. For heterogeneous Clusters with two or more wireless technologies, there may be only one Node that implements multiple wireless technologies. Hence, it is important that such Nodes retransmit and that is also something PFS is expected to do better than CBB due to its self-pruning aspects. In CBB, there cannot be any such guarantees.

Regarding the very high reachability in our measurements, it is important to remember that we used a matrix topology. Such a network poses no challenge to any flooding protocol due to its homogeneous and regular structure. In random networks, extremely inhomogeneous networks, or mobile networks, reachability becomes a challenge. Unfortunately, we cannot easily setup such scenarios in a fair and reproducible way. In those cases, as well as cases related to scalability, we have to resort to simulation.

5.6 Simulation Results

As long as good accuracy can be achieved, simulation will remain an important tool for performance analysis of flooding protocols. In this section, we attempt to simulate CBB and PFS and compare the simulation results with the experimental results obtained in Section 5.5. We chose the ns-2 simulator [159] (version 2.27), which is a frequently used tool for simula-

Table 5.3: Simulation Parameters

Protocol	Simulation Parameter	Value
All	Number of nodes	50
All	Transmission data rate	2 MBit/s
All	Transmission range	100 m
All	Simulation time	100 s
All	Flooding rate	2 packets/s
All	Flooding message payload	64 bytes
CBB	Threshold	3
CBB	T_{max}	30 ms
PFS	Delay parameter (D)	12 ms
PFS	MAX_SLOTS	6
PFS	EARLY_P	10.0
PFS	Hello protocol	Standard[4]
PFS	Hello message interval	1 s

tions of wireless protocols, including flooding protocols. The physical radio characteristics of ns-2 were chosen to approximate the Lucent WaveLAN direct sequence spread spectrum radio. The distributed coordination function (DCF) of IEEE 802.11 [82] was used as the MAC protocol. The RTS/CTS feature was not used since we only use broadcasting. IPv4 is assumed in the simulations, so each node's address is four bytes.

Each node was placed either randomly or in a grid topology within a rectangular (ratio 4:9) and flat area. In the grid topology, the nodes were lined up in a matrix in the same way as in the test bed experiments earlier. The distance between neighboring nodes were altered to achieve different network densities. We used the same way of measuring the network density (formula 5.2) so that results could be properly compared between the simulations and the experiments. We also tested random topologies, where nodes were uniformly distributed in a rectangular area of various sizes, but always with a ratio of 4:9 between the sides.

In each experiment, 200 flooding messages were generated from random nodes at random times. Each simulation was repeated 20 times with the 99 % confident interval calculated and shown in the graphs. Table 5.3 summarizes

[4] Due to the perfect radio model used in ns-2, there is no difference between the three hello protocols

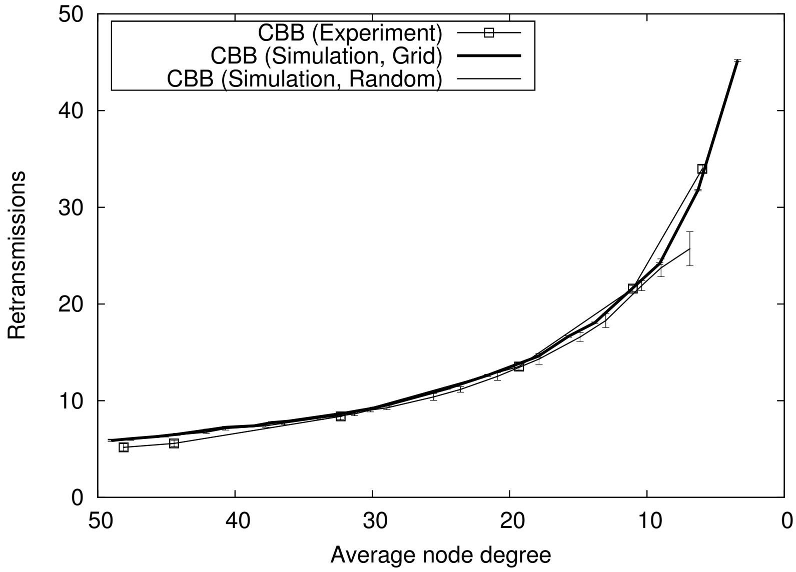


Figure 5.15: Experiment and simulation comparison for CBB

the parameters used in the simulations. To get comparable results with the experiments, we kept the parameters unchanged whenever possible. However, due to the different wireless technologies used in the simulator, CBB's T_{max} and PFS's delay parameter (D) had to be adjusted to the new environment. These values were obtained by a series of simulations, followed by the selection of the values that produced the closest results to the experimental results.

Figure 5.15 shows the results for CBB. In terms of retransmissions, we can see that the grid topology simulation corresponds almost perfectly to the experimental results. Further, switching from a grid topology to a random node topology does not affect the results. The same holds for reachability (not shown), which is very high in all three cases. Concerning the delay, we can see in Figure 5.17 that a similar curve is obtained as in the experiments. However, due to the different wireless technologies, and especially the different transmission rates, the delays differ. From the figure, it appears to be roughly by a factor of five (experiment delay is shown on the left scale, while simulation delay is shown on the right scale).

In Figure 5.16, a similar comparison is shown for PFS. Unfortunately, we cannot see an as good correlation between the simulation results and the experimental results as we did for CBB. It seems that the simulation underestimates the retransmissions of PFS when the network becomes sparser. This may be due to the perfect radio propagation model used by the simulator in which a link is either perfect or non-existent. In the experimental research earlier, we could see the frequent occurrence of links with medium quality. This will have the effect that PFS in the simulation environment will have near perfect neighborhood information, while in a real network, there will be

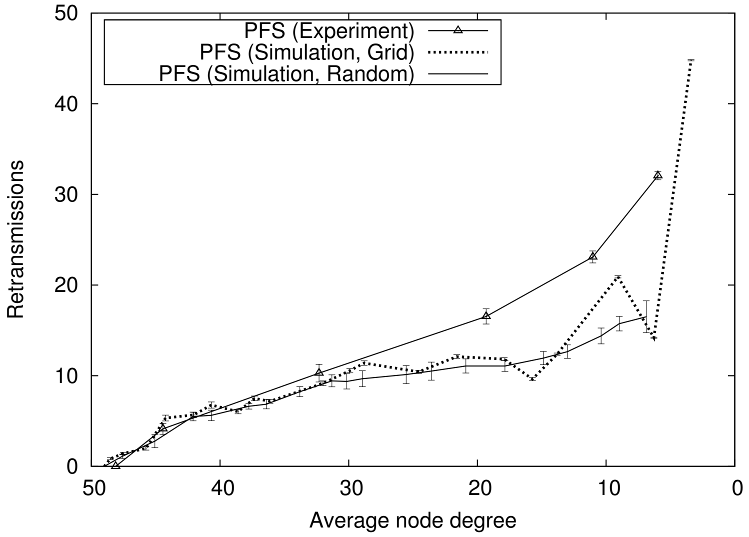


Figure 5.16: Experiment and simulation comparison for PFS

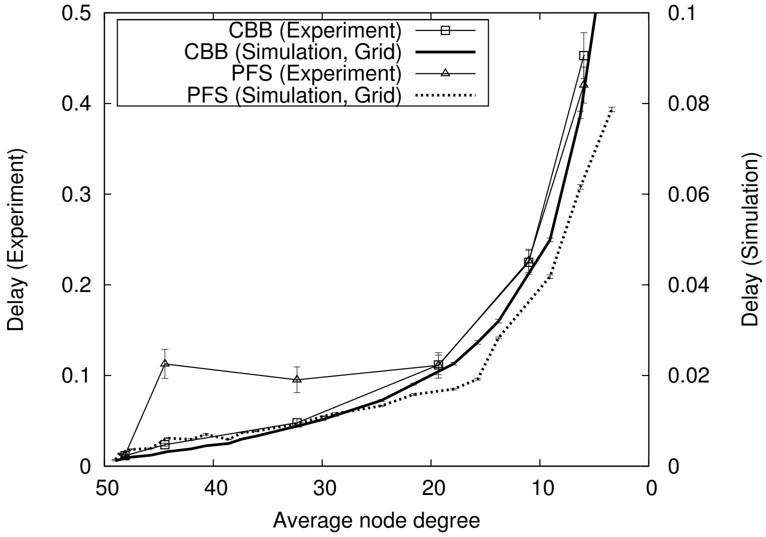


Figure 5.17: Experiment and simulation delay comparison

significant amount of errors. PFS will, because of this, over-perform in the simulator. CBB remains unaffected, since it does not rely on neighborhood information. The delay results for PFS are shown in Figure 5.17. Also from these results, we see a mismatch between the experimental results and the simulation results for PFS. This time it concerns the delay in almost fully connected networks.

The conclusion of this work must be that we need to improve the radio propagation model used in the simulations to better reflect the real situations that can be expected. Hence, we have once again demonstrated the shortcomings of relying on simulations when comparing protocol performance (see

[29]) and the importance of doing measurements in real wireless networks.

To continue this simulation work, we propose to first identify a better radio propagation model and then again verify the simulations against test bed measurements. One approach could be to take our test bed measurement results from the network density measurements, where the delivery ratio is known for every possible link in the network, and build a network based on that. Hopefully, that should give more realistic results.

5.7 The Prototype

We also wanted to know how flooding can be applied to intra-Cluster communication. Therefore, we extended our PN prototype from Chapter 4 with all the four flooding protocols we tested. Which flooding protocol to use was inserted as a parameter in the flooding message itself and hence could be selected by the initiating Node per packet using a configuration option. Hence, it is possible to use different flooding protocols for every packet.

The flooding algorithms were implemented in ppand itself. To allow any application to use optimized flooding without modifications, we let packets with an intra-Cluster multicast destination addresses with a prefix of FF13::/16 be treated as flooding packets by our flooding code. Packets with other multicast destination addresses use the normal link local multicast mechanism only. The total extra code required for implementing all four flooding algorithms measured only 650 lines of code (LoC).

To implement Cluster-wide flooding, we needed to make some small additions to the intra-Cluster data traffic message introduced in Figure 4.6. The new format includes a message identifier, a neighbor Node list, and the number of Nodes in that list. The new message format is shown in Figure 5.18 with the new fields mark with gray. The type of flooding protocol to be used is determined by the message type value, which can be set to either one hop (unicast traffic), Blind Flooding, CBB, or PFS. To use CB-PFS, we use the PFS message type and then set the threshold value to something lower than unlimited.

The system distinguishes packets from each other by using the Msg ID field. A packet is uniquely identified by the PN ID, Node ID, and Msg ID. To make sure a receiving Node does not process a packet twice, it keeps a list of recently received packets and compares new packets it receives to packets in the list. If a match is found in the list, a received packet is handled as a duplicate, otherwise as a new flooding packet.

Another option to duplicate packet detection is calculating a hash value of the packet [128] and only use an extra field when two different flooding packets accidently have the same hash value. The benefit of this option is reduced communication overhead. However, this comes at extra computational costs.

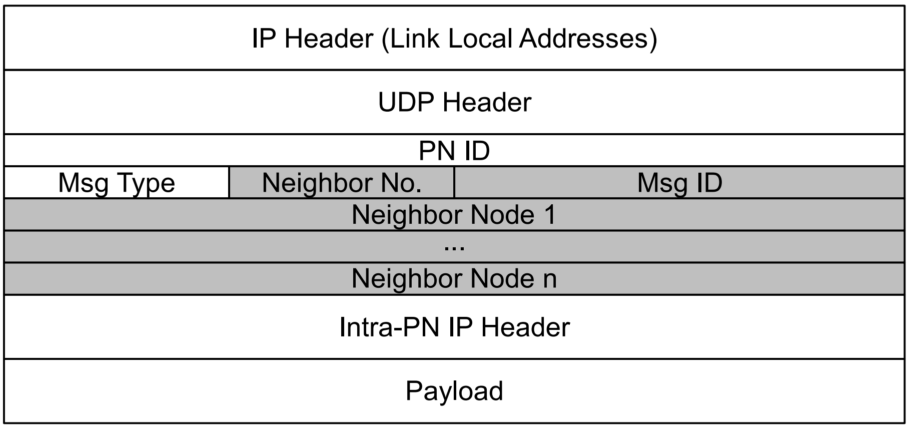


Figure 5.18: Intra-Cluster data traffic message with flooding support

When using PFS or CB-PFS, the neighbor information can come from ppand's hello protocol. However, to implement the lenient and the strict hello protocols, we needed to make some minor modifications. The reception time of the last three hello packets were kept for each neighbor and that was the information we used to determine which node is a neighbor. At the same time, we kept neighbors in the neighbor table longer; until no hello packets were received for some time (50 seconds in our implementation). In the next chapter, we will also update the hello protocol used for unicast traffic.

5.8 Summary

In this chapter, we introduced optimized flooding for intra-Cluster broadcasting. Several flooding protocols were introduced and discussed. We also introduced a new flooding protocol called Prioritized Flooding with Self-Pruning (PFS) and then compared it in a real wireless multi-hop network test bed with two existing flooding protocols: Blind Flooding and Counter-Based Broadcasting (CBB). Blind Flooding is the simplest protocol, but also creates the most amount of overhead. CBB limits the retransmissions of the flooding messages by listening on how often a message already has been retransmitted. PFS relies on neighborhood information collected from a hello protocol to determine the most optimal nodes to retransmit. We also proposed to combine PFS and CBB into a protocol we call Counter-Based PFS (CB-PFS).

We found that while all four protocols achieve very high reachability, CBB, PFS, and CB-PFS create much less overhead than Blind Flooding. However, Blind Flooding was the fastest protocol to reach all nodes in the network. When comparing CBB and PFS, we found that CBB was better in some circumstances, while PFS was better in other circumstances. For instance, PFS was better in fully or almost fully connected networks and very sparse networks—characteristics typical to Clusters. Hence, we believe PFS is more suitable for intra-Cluster broadcasting. Regarding CB-PFS, we could conclude that it created less overhead than both CBB and PFS in all situations. However, we could see a small tendency to lower reachability for CB-PFS. Hence, if total reachability is required, PFS is the best option for Cluster-wide broadcasting; otherwise CB-PFS may prove a better candidate.

The comparison was made in a real network rather than in a simulator due to difficulty of creating an accurate simulation environment. This, we demonstrated by comparing the results of our measurements with simulations done in one of the most popular simulators for wireless communication: ns-2. In the case of PFS, the results from the measurements and the simulations differed significantly and this may be due to the perfect channel conditions assumed by the simulator.

Finally, we implemented all four flooding protocols in our PN prototype that we introduced in the previous chapter. Practical implementation aspects were discussed, such as addressing and how to distinguish new flooding packets from retransmissions. How we used the existing hello protocol for flooding purposes was also discussed. In the end, we concluded that the implementation for Cluster-wide broadcasting using optimized flooding was very small.

Chapter 6

Unicast Routing in Clusters

Most of the traffic within a Cluster will have only one destination. The routing of such traffic is called unicast routing and is very important since the quality of unicast routing directly affects the quality of the end-user applications. Therefore, we look at unicast routing within a Cluster in this chapter. There are plenty of unicast routing protocols for networks, such as Clusters, but they only provide a part of the solution. Almost none of the well-known routing protocols consider how to identify the best path, but instead focus on signaling and signaling overhead. To make unicast routing a success, we do not only need an efficient and accurate routing protocol, but also a good approach to correctly assess the quality of the wireless links and thereby be able to identify the paths with the highest quality.

In most cases, the link quality is assessed based on the exchange of hello packets. Even though several hello packets were successfully exchanged, a link may still not be good enough to carry data traffic. Wireless links have a quality which varies in time to such a degree that it is often not clear whether a link is suitable for data transmission or not; they do not exhibit an on-off behavior. Earlier experiments [44][46] have shown that links with 40 % to 80 % delivery ratio are common in both indoor and outdoor environments. Hence, a better assessment of the link quality is necessary to achieve satisfactory communication within a Cluster. Since a Cluster may use very different types of links and the devices may be of varying capabilities, there must be such mechanisms for each potential wireless technology as well as a mechanism to compare link quality assessments among different technologies. For this to work, we inevitably need cross-layer information.

This chapter is structured as follows. Section 6.1 introduces the most important requirements for unicast routing. Related work is covered in Section 6.2. In Section 6.3, we look at various possibilities to improve the link quality assessment and thereby improve the selection of paths. In Section 6.4, we test the various approaches in our test bed and report the results, which indicate improved routing performance. Some issues related to the routing protocol when using our link quality assessment proposals are introduced in Section 6.5. In Section 6.6, we describe the modifications we did to our prototype to implement all this. A chapter summary is given in Section 6.7

6.1 Requirements for Routing in Clusters

To achieve satisfactory unicast routing and communication within a Cluster, we need a routing protocol that fulfills the following requirements:

1. It must allow unicast routing between any pair of Nodes in the Cluster. This allows any Node in the Cluster to communicate with any of the other Nodes in the same Cluster.

2. The paths selected by the routing protocol must have the quality required by the applications and the user's preferences. The routing protocol should be able to identify the possible paths and then select the path with the most suitable quality. This is especially important for real-time applications. The criteria for the quality of a path are typically low packet loss, low packet delay, and high throughput. Further, energy efficiency, stability, and reliability can also be considered as important criteria. Since applications and typical usage patterns may change, the path selection should be able to adapt automatically.

3. The identification and setup of paths must be timely. How fast the path setup should be depends on the application. Obviously a shorter setup time is better.

4. Routes should adapt to changing conditions. Due to mobility and dynamics in the Clusters as well as changing network conditions, paths may break or change quality. The routing protocol must be able to detect such changes, identify better paths, and gracefully switch to those paths.

5. The routing protocol should create as little overhead as possible. Cluster Nodes are predominantly mobile and hence less powerful; moreover they are likely to be battery powered, which implies that the routing protocol must use its available resources cautiously. It should minimize the amount of signaling traffic and computation.

6. The routing protocol should be robust against errors and security attacks. External as well as internal disturbances caused unintentionally or intentionally should be dealt with in an appropriate way. The protocol should still be able to meet its requirements, while errors or attacks occur.

Most of these requirements are contradictory. This means that a balance between them must be achieved. To achieve a satisfactory level of overhead, only one single routing protocol using no more than one path between node pairs can be used within a Cluster simultaneously. This means that the routing protocol and the paths it selects must sufficiently satisfy all applications and all requirements. Unfortunately, what is sufficient and what the balance should be are very difficult to quantify. In fact, this may change over time as new applications arrive and new wireless technologies are deployed. Hence, we also propose that the intra-Cluster routing protocol is reconfigurable and that there are mechanisms to automatically reconfigure the Cluster to cope with new situations.

6.2 Related Work

The extensive work on routing protocols is obviously of relevance. However, most research on routing protocols typically only considers the signaling aspects of routing, such as how routes are discovered and communicated throughout the network. Some of the popular protocols are introduced in Section 6.2.1. In Section 6.2.2, we also look at methods proposed for identifying and selecting the highest quality path; in particular the choice of route metric. This is equally important as a poor choice will lead to poor end-to-end quality for the applications. In particular, the shortest path frequently may contain very long links with poor quality. Instead another path with one or two extra hops may provide a better end-to-end quality.

6.2.1 Routing protocols

Over the years, many routing protocols have been proposed for all kinds of networks. Since Clusters are dynamic and mobile, we need a protocol that can cope with frequent topology changes, which means that we need to look at the ad hoc routing protocols. Also in this class, several protocols have been proposed. We will briefly introduce the most important ones here, but there are many more. See [5] for a more complete overview of ad hoc routing protocols.

Dynamic Source Routing (DSR) [102][104] was one of the first protocols to specifically address mobile ad hoc networks. A node should not need to keep any routing information and no route updating information should be distributed if not used. Instead, it is up to the sending node to discover and maintain a route to a destination on demand. The path is sent together with the data, in the data packet header, so that all intermediate nodes know where to forward the packet. To discover the path to a destination, a route request packet is broadcasted using blind flooding through the network. If the destination node receives the route request, it answers by sending a reply packet back via the reverse path. At this point, the sending node can start to send data along that path by putting each hop of the selected route into the header of the data packets. If a route becomes invalid, a route error packet is sent back to the sender by one of the intermediate nodes. The sender then needs to find a new route to the destination by broadcasting a new route request message.

There are a few additional features that can be implemented that tries to improve the efficiency of the DSR protocol, such as replying to a route request using cached and overheard routing information, packet salvaging using alternative routes when the selected route fails, and automatic route shortening. All are described in [104].

Ad Hoc On-Demand Distance Vector (AODV) [176] is similar to DSR, but the path is not transmitted in the data packets. Instead, the intermediate nodes have to remember the path. This means that each node in the network only must remember all active paths, which is much less than all possible paths in the network. The route request handling in AODV can be made more efficient than in DSR, since not only the destination node may answer the route request packet. Any node that has an active path to the destination node may answer the route request. Dynamic MANET On-demand Routing (DYMO) [32] is the continuation of AODV towards standardization. The main enhancements are a clearer specification and the use of the common packet format for ad hoc networks [38].

DSR, AODV, and DYMO are all reactive routing protocols. The main drawback is the delay introduced due to the route setup. Common PN applications with many short sessions will suffer the most from this delay. The other option is proactive table-driven routing protocols. Destination-Sequenced Distance-Vector Routing (DSDV) [175] was one of the first proactive ad hoc routing protocols. It is a distance-vector routing protocol suitable for high mobility. With a new way of using sequence numbers, it can detect and avoid routing loops even when the mobility is very high. The main problem of DSDV is its poor scalability. As the number of nodes and the mobility increase, the overhead quickly becomes too burdensome.

A more scalable approach to proactive routing in mobile networks is Optimized Link State Routing Protocol (OLSR) [37]. OLSR is a link-state routing protocol that builds a backbone and only communicates link states belonging to the backbone or links connecting other nodes to the backbone. Most links in the network are ignored and this, in combination with the use of the optimized flooding protocol MPR to broadcast link states, makes OLSR much more scalable. Still, it provides near optimal routing between all nodes in the network. OLSRv2 [40] is a newer version of OLSR that uses the common packet format for ad hoc networks [38], but otherwise uses the same mechanisms and algorithms.

Both reactive and proactive routing protocols have their pros and cons. Depending on the network condition (mobility, number of active paths, path durations, etc), different routing protocols will be the best [69][77]. To address this issue of adaptability, hybrid protocols, which combine the reactive and the proactive approaches, were proposed. One hybrid protocol is the Zone Routing Protocol (ZRP) [68][69]. In ZRP, each node proactively maintains topology information within a local neighborhood, called a node's zone. The zone is defined as all nodes not more than a certain number of hops away. If a node wishes to communicate with a node outside its zone, it needs to discover a path in a similar way to a reactive routing protocol. However, using the local neighborhood information, the flooding of the route request can be optimized in the same way as mentioned in Chapter 5 and this results in less created overhead.

Another approach to adaptive routing is the multi-mode approach taken by the Adaptive Multi-Mode Routing Protocol [74][77]. In this protocol, a node may operate in either proactive or reactive mode. Depending on the network condition, each node may switch to what it believes is the best mode. The protocol is designed in such a way that all modes are compatible with each other, allowing the network to adapt to the most efficient mode. Furthermore, it is possible for the network to use different modes in different parts of the network when the conditions are different. Through simulations, it was shown how the protocol can adapt better to new network conditions and operate efficient in varying network conditions.

There are many more routing protocols, but the ones we discussed have become the most popular for various reasons. Since implementations of all but the most popular routing protocols are rare, we are more or less forced to pick one of these or implement the routing software from scratch. Due to time constraints, we selected OLSR, which after all is a very competent protocol that meets our requirements. It is a reactive protocol that can adapt to changes and creates relatively little overhead. The purpose of this thesis is not to develop or compare ad hoc routing protocols, but merely to demonstrate the possibility of unicast Cluster routing.

There are many more routing protocols, but the ones we discussed have become the most popular for various reasons. For our experiments, we selected OLSR, for which there is a suitable implementation available. It is a proactive protocol that can adapt to changes, but still creates relatively little overhead and hence can support most of our requirements. The purpose of this thesis is not to develop or compare ad hoc routing protocols, but merely to demonstrate the possibility of unicast Cluster routing.

6.2.2 Routing metrics

Early ad hoc network protocols, such as AODV [176] and DSR [102], try to find the shortest path between the source and destination. They mainly use hello packets to detect neighbors and links. However, this does not lead to the best end-to-end performance. Paths with the minimum number of hops often may include very long links with poor quality and such links usually create more retransmissions and use a reduced throughput if multi-rate is supported on such links. Research has shown that there is room for improvement [46].

Link layer techniques, such as forward error correction (FEC), rate control, transmission power adjustments, and retransmissions (also known as ARQ), help improve the quality of a particular link. However, there is always a limit to the impairments that can be handled by such techniques. When the quality of a link drops, more retransmissions are required which means lower throughput as well as longer delays. FEC must use more redundant bits to recover transmission errors, which also costs throughput. At some point, no link layer mechanism is able to transport a single bit; the wireless channel is simply too bad. Hence, alternative paths that avoid bad links are preferred, even if the paths are longer in terms of number of hops.

One of the simplest ideas for improvement is to exclude links with bad performance. For instance, links with more than 50 % packet loss at the lowest transmission rate and highest transmission power are excluded. Using the remaining links, the routing protocol finds the shortest path as usual. The drawback with this approach is the balance one needs between allowing bad links and partitioning the network. A too high packet loss threshold breaks the connectivity of the network and a too low threshold allows too many bad links to be used.

A more promising approach is to minimize the number of transmissions required to reach the destination [47] instead of minimizing the number of hops. This is referred to as estimated transmission count (ETX). The link ETX is the expected number of retransmissions required to successfully transmit a packet over a link and is the inverse of the packet delivery ratio. The routing protocol then needs to find the path with the lowest ETX. This will lead to the one with the least amount of transmissions, including retransmissions, being used to deliver each packet to its destination; this will increase throughput and minimize delay. ETX will find paths with a good balance between the number of hops and the number of retransmissions per hop.

Another promising alternative is to minimize the total medium time required to reach the destination, which is known as estimated transmission time (ETT) [16][22]. In [16], the metric is actually referred to as medium time metric (MTM), but we will use the term ETT since both proposals are almost identical. ETT is actually similar to ETX except that it also considers the link rate. If all links use the same rate, ETX and ETT are in fact the same. Hence, in networks with multiple link-layer technologies using different transmission rates or when using multi-rate link-layer technologies, ETT is better. ETT can find a good balance between link rates, retransmissions, and number of hops.

Both ETX and ETT find paths with expected minimum required transmissions including expected retransmissions or minimum required "channel time". Both will work well in moderate sized Clusters, since most Nodes are within contention range. Only very large ad hoc networks can benefit from approaches where one tries to route around areas with a lot of contention

[72]. However, if different links can use different non-overlapping channels, it might be advantageous to use Weighted Cumulative ETT (WCETT) [53]. WCETT also accounts for the interference between the links. It chooses as much as possible links that use channels that do not interfere with each other in order to maximize the end-to-end throughput.

6.3 Link Quality Assessment

From the discussion in the previous section we can conclude that ETX will be the best option for Clusters using single rate wireless technologies, ETT if using multi-rate technologies, and WCETT if using multiple channels. However, for routing protocols based on any of these approaches to operate well, very good and up-to-date information about the quality of candidate links is crucial. This includes both the current rate and expected packet loss. Furthermore, this information needs to be up-to-date and changes must be detected immediately so that a switch to an alternative path can be quick enough not to disturb the applications. The link quality information must be collected without consuming too many resources, such as energy, computational power and the valuable wireless bandwidth. This functionality, we refer to as *link quality assessment* (LQA).

The goal of LQA is to provide information for maximizing the quality of the end-to-end links within the Cluster. For most applications, this means to achieve as low a packet loss and delay as possible and to maximize throughput. Hence, the routing protocol must identify the best path for each session while not inflicting too much overhead. This is made quite difficult, because typical Clusters contain several very different communication technologies and Nodes with varying capabilities. At the same time, Clusters may be very dynamic with constantly changing conditions.

The LQA mechanism must feed the routing protocol with as accurate and up-to-date information as possible about the quality of available links. The main target is to predict the quality of a link and preferably also predict the link quality trend. It is the task of the routing protocol to act on these changes. This can involve distributing topology update messages throughout the network to make relevant nodes aware of the change. While this certainly improves the awareness of the quality of the links in the network, it also creates tremendous overhead. Therefore, mechanisms are needed that can filter out insignificant changes and frequently changing links, but still allow big or important changes to be propagated. This is an important task for the routing protocol, which we will cover in Section 6.5. For the LQA function, we want a solution that is able to quickly and accurately detect link changes or even anticipate these.

Several methods can be considered to assess the quality of the link. All of them include testing the link using hello packets or passively observing the transmission of ongoing data traffic. In the remaining of this section, we will study the most promising techniques for LQA.

6.3.1 Hello packets

A simple way to predict link quality is to continuously and periodically generate hello packets. If the hello packets are received at a neighboring node, then there is a link. The packet loss can be estimated by transmitting a known number of hello packets over a certain time and observe how many of them arrive, e.g., counting how many of the last 10 hello packets have been received. Numerous protocols [102][176][37][40] and implementations use this method. Common is to broadcast a hello packet every X seconds, where X may vary between 1 second and up to perhaps 10 seconds. By using broadcasting, one transmission can be used to detect the link quality with all neighbors at once and also to discover new neighbors. The link layer usually does not use retransmissions for broadcast traffic and hence the real packet loss ratio before retransmissions can be determined.

In most implementations, the window size for counting the number of hello packets is 10. Suppose the hello packet interval is 1 second, then it takes 10 seconds for a node to fully learn the quality of the link to a new neighbor. In the same way, it takes this mechanism 10 seconds to fully learn the new quality of a link when it changes. For Clusters, which are dynamic in nature, this mechanism reacts too slowly to link changes. If a link suddenly breaks or changes quality, it typically takes several seconds before the routing layer detects this and takes action. When this happens frequently, it causes detectable quality drops and severely degrades the performance of real-time applications running on top of these networks. Hence, the packet delivery ratio measured by hello packets is not accurate and fast enough for Clusters.

Increasing the hello packet interval will make the prediction faster and more accurate, but will also significantly increase the overhead. Hence, it is not a feasible solution. To improve the responsiveness of this measurement, we can give a higher weight to more recently received (or lost) hello packets. Of course, a higher weight on more present packets will lead to faster adaptation to link changes but it will also lead to a less accurate prediction. The last hello packet influences the prediction too much, which makes it a less stable predictor.

The fact that hello packets can be lost due to both bad wireless channel conditions and collisions makes it even more difficult to use. It is very difficult to know the exact cause of a packet loss. However, collisions are more likely to happen in saturated parts of the network or will affect all links to the same extent. Hence, if the collision rate can be controlled, this will not pose a serious problem.

Another problem when using hello packets that applies to many link layer technologies, including IEEE 802.11, is the inaccuracy caused by the data rate and packet size differences between hello packets and data packets. Hello packets are typically smaller than data packets and are always sent using broadcasting. Broadcast packets in the IEEE 802.11 protocol family are sent using one of the lowest data rates in order to be backwards compatible (2 Mbps in our test bed). These two differences will cause the delivery ratio measured by broadcasted hello packets to differ from the packet delivery ratio when using the same link for data traffic [31][11].

6.3.2 Signal strength

The most commonly used cross-layer information for link quality assessment is either the received signal strength indication (RSSI) or the signal to noise ratio (SNR). For example in [35], RSSI is used for gateway selection; the gateway with the strongest signal is selected. Another example is signal stability-based adaptive routing (SSA) [55] in which signal strength is used for route selection in an ad hoc network. The IST 6HOP project [4] implemented a multi-hop ad hoc networking test bed and used signal strength as a routing criterion in one of their setups.

In general, the success of using signal strength for LQA or as a routing metric has not been very promising. The reason is the weak correlation between signal strength and packet loss. Several studies of this have been done. See, for instance, [44] or [8]. Figure 6.1 shows an example plot of signal strength versus delivery ratio for IEEE 802.11 at 2 Mbps (using broadcast). It is based on an experiment in which we sent 10 packets/s with a size of 36 Bytes, excluding Ethernet headers. Each point is an average of the signal strength and packet loss over 10 seconds (i.e., 100 packets). The conclusion that can be drawn is that good signal strength usually means good packet delivery ratio, but when the signal strength drops below a certain threshold, it becomes very difficult to predict the delivery ratio. It can be anything from perfect to non-existent. Unfortunately, that threshold is so high that very few links are above. Hence, we can extract very little information from the signal strength.

Another problem with signal strength is its fluctuating behavior. Shadowing and fading wireless channels cause the signal strength to vary even in a controlled and stable environment. Figure 6.2 shows the result of one node constantly transmitting 10 packets/s to another node. The received signal strength of the receiving node is given in the graph. Despite this being a stable indoor environment, we see serious fluctuations even between consecutive packets. It is clear that we need to smooth out these short-term fluctuations. For this, an exponentially weighted moving average (EWMA) can be used:

$$E_{t+1} = \alpha \cdot S_t + (1 - \alpha) \cdot E_t \tag{6.1}$$

In the formula, S_t is the signal strength in dBm of the last received packet, E_t is the previous estimation, and E_{t+1} is the new estimation. α is a constant

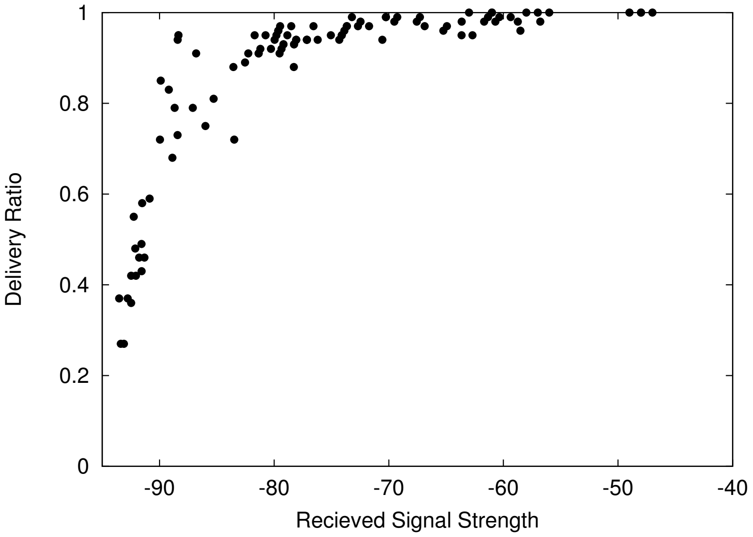


Figure 6.1: Received signal strength vs. packet delivery ratio

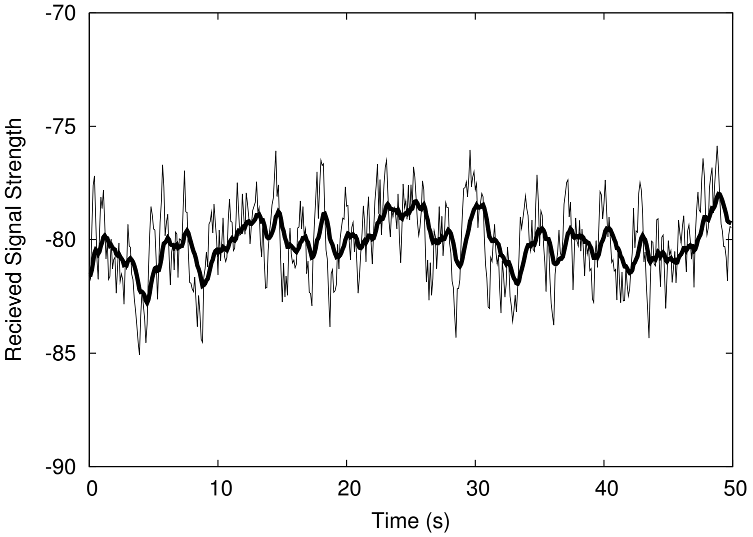


Figure 6.2: Received signal strength over time

that determines the amount of weight assigned to the last measurement. The reason why we prefer a EWMA is its ability to both smoothen out fluctuations and to quickly adapt to new circumstances. A normal moving average weights historical values equally with present values and cause a slower change when the quality shifts.

The difficulty of using a EWMA with random values, such as signal strength, lies in the selection of α. A too small α leads to too fluctuating estimations and a too big α leads to too slow adaptation. A balance is needed that depends on the frequency of the measurements, the amount of fluctuations in the measurements, and the need of fast adaptations. The smoothed estimation that is shown in Figure 6.2 uses $\alpha = 0.1$, which seems right for that scenario.

The last problem with using signal strength is that we only know the signal strength associated with received packets. Hence, the estimation will not be very accurate if there are many packet losses since we do not know the signal strength associated with the lost packets. To account for this, it is better to assume that a lost hello packet has the lowest receivable signal strength instead of ignoring it. In Section 6.4.1, we will demonstrate the improvements this scheme brings.

6.3.3 Data packet retranmissions

When a link is carrying data traffic, we can gather even more information about its current quality. Hello packets are only sent infrequently, which means that the delivery ratio deduced from hello packets only has a poor accuracy and react slowly to changes. If a link carries a lot of data packets, then the delivery ratio of the data packets can also be used to perform LQA. However, there are three problems with this approach:

1. Most wireless technologies provide retransmissions for the delivery of data packets and we need the packet delivery ratio including all transmissions and retransmissions. That is, we need the packet delivery ratio before any retransmission scheme. For many standard wireless cards and their drivers, this information is not readily available. However, there is nothing fundamental that should stop us from retrieving this information. In fact, we succeeded in obtaining this information from the Madwifi open source driver [129] for Linux in combination with a 3Com OfficeConnect Wireless 108Mb 11g PC Card [1]. The retransmission feedback was implemented as two counters for each known neighbor node. One counter is increased for each transmission or retransmission, while the other counter is increased for each successfully received acknowledgment. By reading these counters periodically (e.g., every 100 ms), we can obtain how many packets were successfully transmitted to that neighbor in the last period and how many (re)transmissions were required to achieve that. A similar approach can be used for other drivers, cards, or even different wireless technologies providing retransmissions.

2. The delivery ratio of the data packets depends on the sizes of the data packets. Larger packets have lower chances of successful delivery than small packets, sometimes much lower. See Section 6.5.2 for further analysis of this phenomenon. However, it is impossible to control the packet sizes that are being used and this adds uncertainty to this LQA method. Because of this, we need to compensate for packet size when calculating the packet delivery ratio. The LQA should assume one

reference packet size and convert the delivery ratio of other packet sizes to the reference packet size. The chosen reference packet size should represent the typical packets transmitted on that link.

3. When there is no or very little data traffic on a link, then this approach cannot be used. It is therefore important to not only use the delivery ratio of data packets. Instead, a combination of data packet delivery ratio, hello packet delivery ratio, and signal strength should be considered. When the number of data packets transmitted on a link becomes smaller, more emphasis should be placed on hello packet delivery ratio and signal strength.

Despite these problems, the data packet delivery ratio provides the best option for LQA, since it measures the actual delivery ratio. It is also for links that are actually used for data transmission that we need the best accuracy and fastest detection of link quality changes. It is more crucial that the routing protocol can quickly react on a sudden degradation of a used link in order to maintain a high end-to-end path quality at all times. To quickly react on a quality improvement of a link is usually not required. Maybe the link will not be used immediately anyway and following the cautious approach of Toh [223], it may not be a good idea to quickly start using it. The chance that the link quality soon degrades again is probably very high.

The data packet delivery ratio can also fluctuate significantly on a short time scale and that is the reason why retransmissions are needed in the first place. To be able to use this information for predicting the link quality, we need to smooth out these fluctuations in a similar way to the smoothening of signal strength fluctuations above. Figure 6.3 shows an example of a degrading link with plenty of data traffic. The sender generated UDP packets at a rate of 50 packets/s and a size of 1000 Bytes, excluding Ethernet headers. One of the nodes moved away from the other with an approximate speed of 0.5 m/s while constantly remaining in line of sight. The experiment was carried out in an indoor office environment with interfering wireless communication, including other WLAN networks.

In Figure 6.3, the gray background shows the actual data packet delivery ratio before retransmissions averaged over 100 ms (around five data packets). The thin black line shows the EWMA of the same data packet delivery ratio using $\alpha = 0.1$. It is clear that an average over only five transmissions causes the data to fluctuate too much, which is virtually useless. The EWMA-smoothed values, on the other hand, fluctuates less and hence are better for assessing the current quality of the link.

6.3.4 Other measures

Hello packet delivery ratio, data packet delivery ratio, and signal strength are not the only measures that can be used for LQA. Other measures could be the

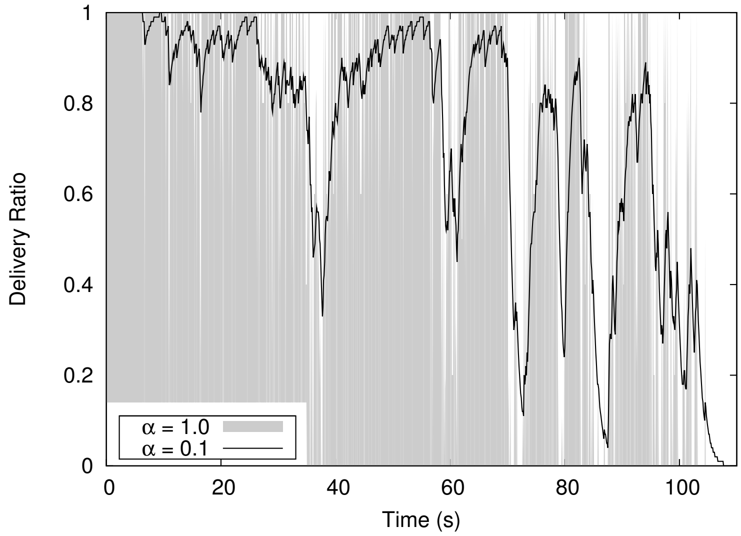


Figure 6.3: Data packet delivery ratio and predictions with different α

number of transmission failures, chip error rate, or overhearing cross traffic between other Nodes. This thesis will not further consider the usefulness of these types of measures, but future research may want to investigate their suitability for LQA.

6.3.5 Bi-directional LQA

It is also very important to point out that a link can have one type of quality in one direction and a completely different quality in the other direction [46][47]. That is, a wireless link can be very asymmetric. Most routing protocols only use one bi-directional metric for a link instead of one metric for each direction. The main reasons for this are lower overhead and the fact that unicast communications involve packets being sent in both directions anyway. Data packets are sent in the forward direction, while acknowledgments are sent in the reverse direction.

Therefore, it is important to realize what information is available at which end of the link. For signal strength, this is not an issue, since signal strength is known to show good symmetry. This has been known since 1896 due to the work of Hendrik Lorentz, which lead to what is known as the Lorentz reciprocity theorem. We could also verify this phenomenon in our test bed. Furthermore, signal strength can also be obtained from acknowledgment packets, so even if the data stream is only going in one direction, we can acquire good signal strength information at both sides.

However, the packet delivery ratio can be very different in the two directions and this needs to be taken into account. Further, the difference in packet size between data packets and acknowledgments is so significant that LQA based on delivery ratio may differ between the two nodes only because of this. It is therefore important that the LQA measures in both directions are combined at some point. This can be done by the routing protocol, by sharing this information using hello packets, or via extra header fields in the data packets. All solutions add overhead, but the differences between the two directions can be so large that it is necessary.

6.3.6 Heterogeneous air interfaces

It is likely that a Node supports more than one type of link layer technology for intra-Cluster communication. In those cases, the Node needs to implement one LQA mechanism for each link layer type. The LQA implementations must be tailored to the characteristics of their respective link layer technology and hence may significantly differ from each other. However, it must be possible to compare LQA results between a particular link layer and the others in a fair way. To achieve this, the LQA results must reflect the behaviors of the link layer techniques.

It is clear that LQA based on ETX is not sufficient, since ETX fails to consider the data rate, which typically is different between different link layer technologies. Furthermore, different link layer technologies may occupy different radio frequencies (or channels). Hence, for Clusters with heterogeneous air interfaces, it is better to base the routing metric on WCETT [53] as discussed in Section 6.2.

The different link layer technologies may also have other characteristics that should influence the selection of one path over another. For example, power consumption may be a factor and hence should be reflected in the LQA. However, this makes it necessary to define a trade-off between power consumption and achievable throughput, but such trade-offs are difficult to define. Instead, it is probably better to make this tunable by the user or have it decided by a context-aware decision system. Such a system may take battery-charge levels and other external factors into account when reconfiguring the used routing metrics. However, this is beyond the purpose and scope of this chapter. Instead, we will focus on finding paths with maximum throughput.

6.4 Experimental Results

A lot of previous research on unicast routing protocols for ad hoc networks has been based on simulations. However, to properly assess the routing protocol, the route selection metric, and LQA, we need to do real network measurements. For this, we used the currently very popular wireless communication standard IEEE 802.11g. A couple of laptops were equipped with 3Com OfficeConnect Wireless 108Mb 11g XJACK PC Cards [1], which implement IEEE 802.11g. The Madwifi driver [129] was modified to be able

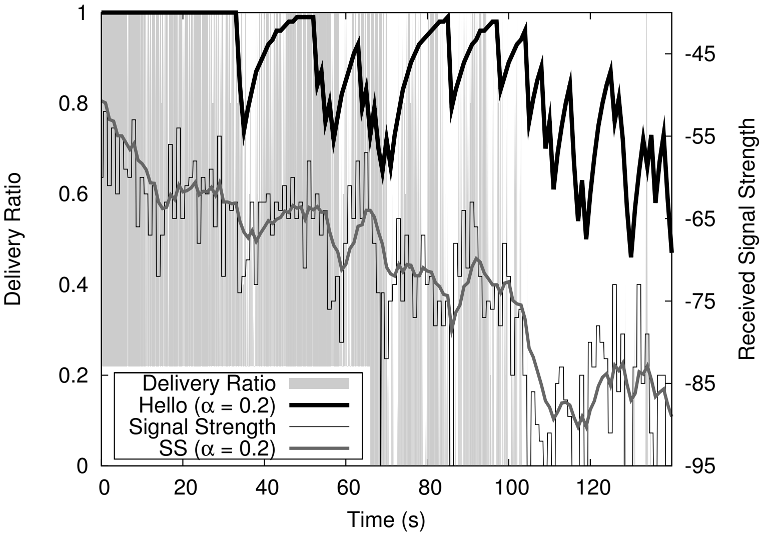


Figure 6.4: Degrading link and its LQAs

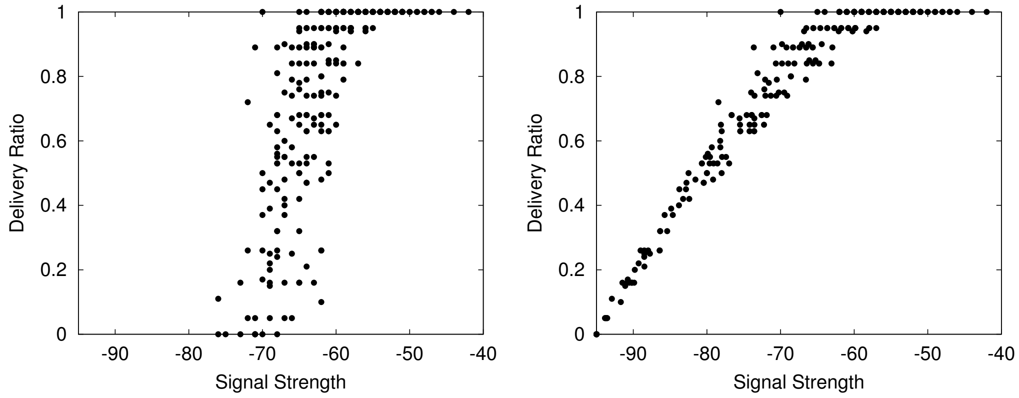
to obtain retransmission information as explained earlier. All the experiments were carried out with the data transmission rate fixed at 54 Mbps and always using full transmission power. The request-to-send (RTS) / clear-to-send (CTS) exchange functionality of the IEEE 802.11 MAC protocol was completely disabled.

We ran a modified version of ppand; our Cluster formation software that we first introduced in Chapter 4. The modified ppand implements LQA by collecting link quality information and computing a link quality metric. Then, the metric is fed to a modified version of the Olsr.org OLSR routing daemon [169] for selection of the best path. More details on how the implementation was done are given in Section 6.6.

6.4.1 LQA experiments

The first set of experiments we conducted aimed at defining an efficient LQA and verifying its accuracy against the actual data packet delivery ratio. Figure 6.4 shows the same degrading scenario as in Figure 6.3, but with different LQA mechanisms. Two laptops were used, where one slowly moved away from the other. The fixed laptop generated 50 data packets of 1000 Bytes per second at a fixed transmission rate of 54 Mbps. At the same time, both laptops generated one hello packet per second of 80 Bytes using broadcasting at a transmission rate of 2 Mbps. This is the most natural setup for multi-hop ad hoc networks based on IEEE 802.11 technology. Unfortunately, this frequently leads to performance problems when the delivery ratio of the hello packets is used as LQA as explained earlier in this chapter.

The black thick line in Figure 6.4 shows the delivery ratio of the hello packets using an EWMA with $\alpha = 0.2$. Compared to the actual data delivery



(a) Received Signal Strength

(b) Processed Signal Strength

Figure 6.5: Signal strength vs. data packet delivery ratio

ratio, illustrated by the gray background graph, it can be seen that the hello packet delivery ratio constantly overestimates the quality of the link. This is due to the smaller packet size and the slower transmission rate being used. Even when no data packets can be delivered, the hello packet delivery ratio indicates a delivery ratio in the range of 10 to 50 %. Furthermore, adaptation to link quality changes is slow. Hence, hello packets alone are not a good approach to LQA.

In Figure 6.4, the black thin line shows the received signal strengths of the hello packets with a lost packet assuming a signal strength of −95 dBm, which is the lowest receivable signal strength for our equipment. It can clearly be seen that the signal strength degrades as the two laptops move apart, but it can also be seen that the signal strength fluctuates. Hence, we need to smooth it out with EWMA, which is shown as the dark gray thick line ($\alpha = 0.2$).

A better approach to LQA, when there are no or very little data packets on the link, is to use a combination of the hello packet delivery ratio and the received signal strength. To investigate the potential of such a combination, we designed an experiment where one laptop constantly sent 1000 Bytes packets to another at a rate of 20 packets/s. The transmitting laptop was moved around to create different scenarios with different packet delivery ratios and different received signal strengths. Figure 6.5 shows the result of this experiment where each measure point contains the average over 1 second (around 20 packets). Figure 6.5(a) shows the average received signal strength versus the average delivery ratio. The correlation is not clear as the signal strength drops below −60 dBm. When we account for the lost packets by assuming a received signal strength of −95 dBm, we get the result of Figure 6.5(b). Suddenly, the correlation becomes much stronger; this hints at a linear relation below the threshold of −60 dBm.

To test the accuracy of a combination of received signal strength and hello packet delivery ratio as a LQA, we used the following formula:

$$LQ_{EST} = \text{MIN}(1,\ C \cdot (1 - \frac{SS_{HR}}{SS_{MIN}}) \cdot LQ_{HR}) \tag{6.2}$$

where LQ_{HR} is the EWMA of the packet delivery ratio, SS_{HR} is the EWMA of the received signal strength, SS_{MIN} is the minimal receivable signal strength, and C is a tunable parameter. There are other ways of defining an estimation function based on hello packet delivery ratio and received signal strength, but this approach was selected in order to identify the applicability of using the combination of hello packet delivery ratio and received signal strength as LQA. Further research may determine if a different formula is better.

In some experiments presented in [242][243], we identified the optimal value of C, which actually depends on the data rate and the packet size. Since we used a rate of 54 Mbps and data packets of 1000 Bytes, the best C was found to be 2.6 with $\alpha = 0.2$ for the EWMA of both the hello packet delivery ratio and the received signal strength. Only signal strength values of hello packets were used. It should also be noticed that these parameters still work well with other packet sizes and transmission rates.

We again used the same setup as in the degrading link experiments in Figure 6.3 and Figure 6.4, but this time with a different scenario. The mobile laptop moved to and from the fixed laptop with each cycle taking about 40 seconds. During each cycle, the link turned from perfect to non-existent and then back to perfect again. The gray background graph in Figure 6.6 shows the data delivery ratio as in the previous figures. The thick black line shows the LQA based on only the hello packet delivery ratio, while the thin black line shows the LQA based on formula (6.2).

LQ_{EST} based on formula (6.2) is far from a perfect LQA, but is certainly a big improvement compared to only using the hello packet delivery ratio. In the following section, we will investigate how this LQA works in a small three-node network.

6.4.2 LQA and path selection

The purpose of the experiment in this section is to verify the suitability of the LQAs proposed earlier in this chapter. For this aim, we used a simple three-node network with one sender and one receiver as shown in Figure 6.7. Two possible paths exist in this network; either the direct path between the sender and the receiver or a multi-hop path via the third node. The sender and the intermediate nodes were kept fixed, while the receiver was moved up and down the hall as indicated in Figure 6.7. The consequence of the movement was that sometimes a direct path was best and sometimes the 2-hop path via the intermediate node was best. The same movement was repeated 10 times and 95 % confidence intervals were computed. Since it

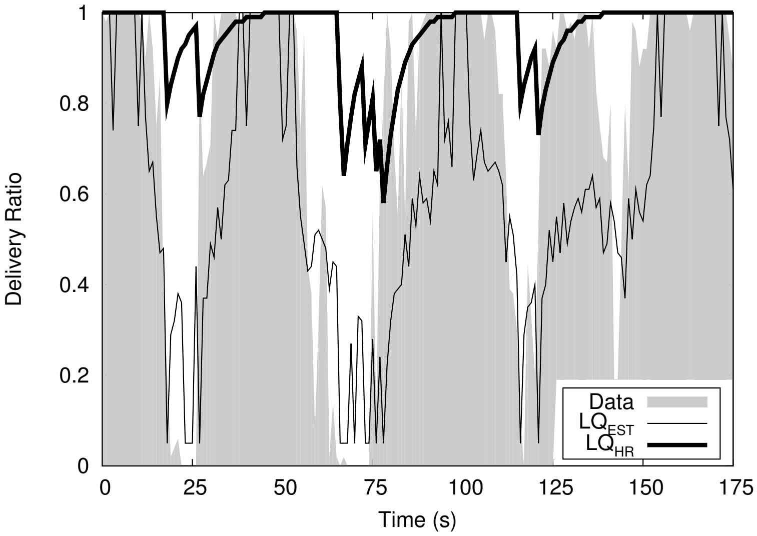


Figure 6.6: LQA using hello packet delivery ratio and signal strength

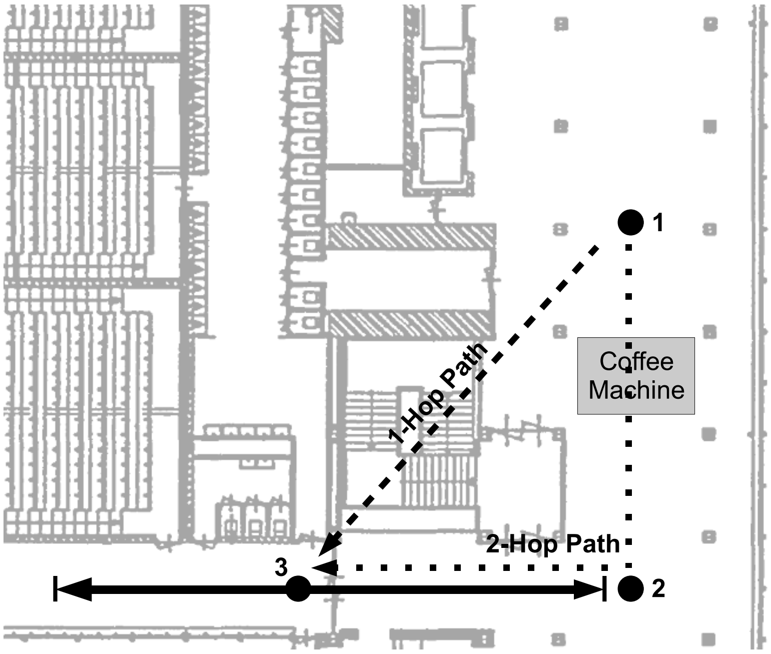


Figure 6.7: Multi-path LQA experiment setup

is impossible to test the LQAs in parallel, each graph stems from different experiments. We did our best to exactly repeat the movements, but this was done by carrying the laptops by hand and hence not perfect. However, by repeating the experiment 10 times, we can still make a fair comparison.

Figure 6.8 shows the average TCP throughout achieved during one movement and its confidence interval for several different LQA mechanisms. We used Linux's standard TCP protocol, which is TCP BIC [235] with selective

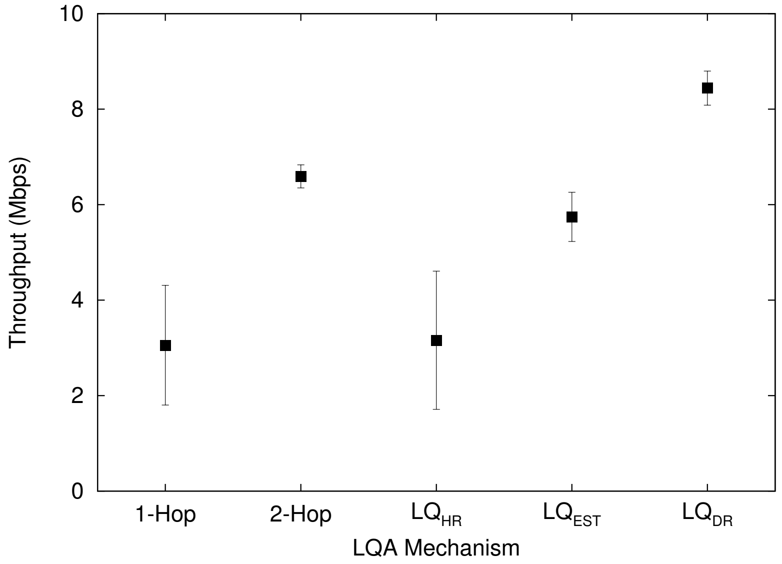


Figure 6.8: Route throughput comparison (TCP)

acknowledgment (SACK) [146]. 1-Hop refers to fixed routing that always used the direct path, while 2-Hop refers to fixed routing that always used the 2-hop path via the intermediate node. To be able to use the data packet delivery ratio (before retransmissions), we had to introduce fake data packets on all possible links. The fake traffic consisted of UDP packets of 1500 Bytes at a fixed rate of 50 packets/s.

LQ_{HR} refers to using only the hello packet delivery ratio with EWMA ($\alpha = 0.2$). As expected, it can be seen that this mechanism always overestimates the link quality and constantly chooses the shorter path. Hence, LQ_{HR} is identical to always using the one hop path. On the other hand, LQ_{EST}, which combines hello packets and received signal strength according to formula (6.2), significantly improves the throughput; it increases from 3.2 to 5.7 Mbps, but is still not better than always using the 2-hop path in this example. However, this is probably the best one can do with the little information used in this LQA. When using information from the data packet delivery ratio, the throughput increases further and beyond also the 2 hops path as indicated by LQ_{DR} in Figure 6.8 (EWMA $\alpha = 0.1$). Unfortunately, this LQA is unrealistic since it requires frequent unicast packets on every possible link.

In Figure 6.9, we show the instantaneous throughput (averaged over 1 second) of one example movement for each LQA mechanism. The thick lines show the throughput for the fixed paths, which shows us that the 2-hop path is best between 1 and 6 seconds and between 32 and 37. Between 7 and 31 seconds, the direct path is best. The remaining three graphs show the achieved throughput for the investigated LQAs. It must be stressed that the five graphs actually do not stem from the same experiment, but five differ-

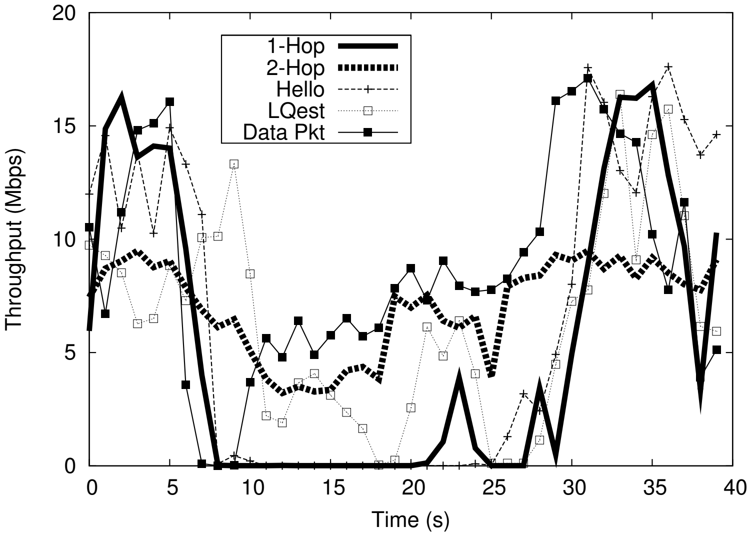


Figure 6.9: Instantaneous route throughput comparison (TCP)

ent experiments. However, we tried to exactly repeat the scenarios as we explained earlier.

With these experiments, we have shown the need for a good LQA and how better LQAs can improve the throughput of a small multi-hop network. Further information of these results can be found in [242].

6.4.3 LQA with rate adaptation

When using LQA in combination with a multi-rate wireless technology and rate adaptation, the expected transmission rate must also be taken into consideration. If this is the case, the routing protocol should instead find a path that minimizes the "channel time" as in ETT [22], which was introduced in Section 6.2.2. This means that the LQA mechanism must return the expected amount of time that the channel will be used to transmit a packet on a potential link, including retransmissions and acknowledgments.

However, rate adaptation, such as SampleRate [21], makes this task a challenge. The "channel time" is nothing more than the transmission rate multiplied with the inverse of the packet delivery ratio. However, it is not uncommon that the rate frequently changes and this must be taken into consideration since it otherwise may cause the LQA result to fluctuate significantly. At the same time, the rate adaptation scheme is not perfect, whereby the current used rate does not necessarily reflect the best current rate. Rate adaptation schemes use mechanisms similar to the LQA approaches in this chapter to decide which rate to use [106]. Initial testing seems to indicate that SampleRate fails to quickly adapt to link quality changes. A new promising rate adaptation scheme is the Channel-Aware Rate Adaptation Algorithm (CHARM) [106], which we have not been able to test. If CHARM or another rate adaptation scheme works well, a promising future direction is to extend our LQA approaches with multi-rate technologies and develop working combinations between LQA and rate adaptation.

6.4.4 LQA with and without data traffic

The results above suggests using data packet delivery ratio (LQ_{DR}) when available or otherwise using LQ_{EST} according to formula (6.2). However, we must be careful when taking this approach and make sure the results of the two approaches are comparable. If the LQA changes because it switches from LQ_{EST} to LQ_{DR}, it may create instability in the network. In the worst case, the routing may start to oscillate, which may happen if the LQA based on LQ_{DR} is lower than LQA based on LQ_{EST}. As soon as a link is being used, it appears like the LQA suddenly drops, which makes the routing protocol look for a better alternative. However, a rerouting away from that link makes its LQA improve and the routing protocol again reconsiders its decision. Because of this, it is better if LQ_{DR} slightly overestimates the link quality compared to LQ_{EST}. However, more research is required to gain more insight in this aspect and fine-tune the solution.

6.5 Routing Protocols for Clusters

There are plenty of routing protocols that can be used for intra-Cluster routing. Only a small fraction was introduced in Section 6.2.1. Several studies that compare the different protocols and their performance have been made, such as [26]. Most of them are based on simulations and indicate that each routing protocol has its own advantages and disadvantages [74]. Thus, one protocol may outperform the others in some scenarios, but not in others. The studies mainly focus on the amount of overhead created and the ability of the protocols to find the shortest path at all times. In [135], a simulation comparison specifically targeted towards intra-Cluster communication is presented. There, two common routing protocols; AODV [176] and OLSR [37] were compared. The same type of conclusion was reached; the best choice of protocol depends on the scenario. Instead of delving deeper into the issue of selecting the best routing protocol, we will focus on how to make best use of the improved LQA in a routing protocol and the issues that arise from that.

6.5.1 LQA updating

The LQA results need to be fed to the routing daemon for further processing. The routing daemon can communicate the LQA values to other nodes in the network and then compare the LQAs of links and identify optimal paths. This means that the LQA values are constantly sent to the routing daemon. Since this is a mechanism internal to a node, the updating can be frequent.

However, the routing protocol cannot distribute those LQA changes to the other nodes at too high a rate. It would create a lot of overhead, which degrades the performance of the Cluster and drains power. At the same time, the distribution of LQA changes can lead to improved paths; this in turn improves the performance and lowers the required power.

A strategy for how and when to distribute LQA changes is required. First, periodic updates are always required. A routing protocol needs to make use of soft states, meaning that every state must be refreshed or otherwise discarded after a timeout. This is required to make sure stale data is removed when networks and nodes disappear without prior warning. In addition to the periodic updates, significant events should preferably trigger immediate updates. The main questions are: how frequent the periodic updates should be and what is a significant event. More research is required to identify a good balance between the overhead created and the improvements to the identified paths.

6.5.2 Different packet sizes

As has been mentioned earlier, packet size influences packet loss, with bigger packets experiencing bigger packet loss on a link. On a wireless link without forward error coding (FEC), [162] showed that the packet loss increases exponentially with the packet size and that the packet loss doubles for each additional 300 Bytes. We decided to verify this behavior in our test bed.

Figure 6.10 shows the result of an experiment where 16 different packet sizes were generated continuously at a high frequency and in random order between two laptops. The packet sizes shown in the graph includes the Ethernet and IP headers. For simplicity, we used 2 Mbps broadcast packets on the wireless link to avoid link layer retransmissions. The measurements were done in 8 different static scenarios with different conditions and different distances. Some scenarios were setup in a typical office environment with several other interferers or in a typical home environment with minimal interference. In each experiment, we generated 200 packets of each packet size using a Poisson arrival process during about 3 minutes and 20 seconds. That meant that we utilized the wireless channel less than 5 % of the time.

For each scenario, we conducted 10 experiments in short succession with which we could calculate the 95 % confidence interval. Each line denotes the packet loss for the different packet sizes for one static scenario as an average of the 10 experiments. We can see that the packet delivery ratio indeed decreases with the packet size. Sometimes this difference can be quite dramatic; going from 46 % packet delivery ratio for 50 Bytes packets down to only 6.5 % for 1500 Bytes packets in one scenario.

From the results in Figure 6.10, we can also see that it is not possible to accurately estimate the packet loss for all packet sizes if we only generate hello packets of one packet size, since we cannot know the slope of the curve.

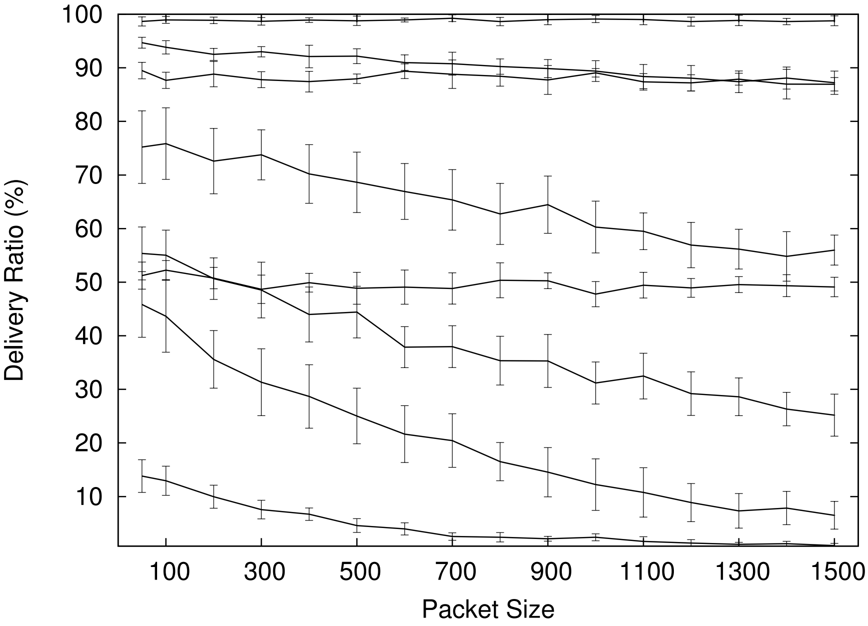


Figure 6.10: Packet size vs. packet loss

Perhaps it is necessary to use hello packets of two different packet sizes in order to accurately estimate the packet loss. Furthermore, we can conclude that the observations from [162] do not correspond with these results. Perhaps this is due to the very good channel conditions (loss rate lower than 1 %) used in [162]. In our measurements, there is no clear exponential relationship between the packet loss and packet size. In fact, there is sometimes a very large packet size independent-related loss (the curve has almost no slope). However, further research is needed in order to better investigate the relationship between packet size and packet loss as well as the impact of interference, which may play a major role here.

The consequence of all this is that the optimal path may be different for small packets (e.g., VoIP packets) than for big packets (e.g., file transfers). It is possible to report the LQA for each packet size (or class of packet size). Unfortunately, a routing protocol cannot afford the extra overhead of maintaining multiple routes between nodes, since the overhead is already daunting. This means that one route has to fit all types of data packets. Hence, a representative packet size should be assumed. To identify the best packet type to represent all data packets in a Cluster is not trivial. It depends on the packet size distribution, which may shift when the usage pattern shifts. The simplest solution is to measure the expected packet size distribution offline in a typical Cluster and identify the best packet size and then fix that size for all Clusters. Future research may investigate whether it is beneficial to have a dynamic mechanism that monitors the packet size distribution within a Cluster and adapts the reference packet size accordingly.

6.5.3 End-to-end quality of intra-Cluster paths

Let us now look at the end-to-end quality that can be expected within a Cluster and the relationship with transport layer protocols. For UDP-based applications, the results given above will give a good indication of the end-to-end quality that can be expected. However, for TCP-based applications, the experienced quality is different due to the way TCP operates. Standard TCP assumes that packet loss is due to congestion and therefore starts to reduce transmission speed when experiencing loss. This works fine in wired networks that have very little non-congestion-related packet loss. However, on a non-congested wireless link, as little as 1.55 % packet loss gives a TCP throughput reduction of more than 17 % [162] while a packet loss of 2.3 % gives a reduction of about 50 % [18]. This is the main reason for most link layer technologies to implement acknowledgments and retransmissions, which significantly lowers the end-to-end packet loss.

Retransmissions, on the other hand, introduce unpredictable delay jitter and this can be a problem for TCP. TCP measures the end-to-end delay and then uses this to accurately and timely detect packet loss. If the delay introduced by link layer retransmissions fluctuates too much, it may cause TCP to believe that packet loss has occurred, which leads to reduced throughput. Furthermore, retransmissions do not completely eliminate packet loss. When the wireless channel becomes saturated, the retransmission scheme will still fail and drop the packet due to busy channel conditions or collisions.

However, TCP can be modified to better cope with these conditions. Several proposals exist that may improve the performance of TCP in wireless networks. Explicit Congestion Notification (ECN) [191] and Explicit Loss Notification (ELN) [50] send special messages to distinguish between packet drops due to congestion and drops due to the imperfect wireless link. TCP Selective acknowledgment (SACK) [146] improves acknowledgments by reporting exactly which segments were lost and with this extra information, the congestion avoidance of TCP can make more intelligent decisions also when multiple losses are detected. TCP Eifel [126] improves the recovery scheme by solving problems caused by spurious timeouts. There is also a whole range of congestion control algorithms that can improve the response to packet losses, such as TCP BIC [235], TCP Vegas [24], and TCP Westwood [145]. The effect of some of these TCP modifications on the performance over wireless links has previously been investigated in [18]. Furthermore, there are also many proposals suggesting cross-layer optimizations between TCP and the lower layers, such as the Snoop protocol [17].

TCP over mobile multi-hop wireless networks has also been studied [34]. The most important difference with fixed networks is the fact that links in a multi-hop wireless network share the same medium. That is, a transmission on one link affects the performance on other links and this makes it difficult for TCP to work properly. In [60], it is shown that a limit to TCP's congestion window size (which determines the amount of packets in transmission) can improve the TCP throughput and a badly selected window size can reduce the performance with about 20 %. Several papers (e.g., [60] [219]) also show that unfairness among TCP connections becomes larger in multi-hop ad hoc networks. To some degree, the fairness problem can be alleviated by using a better medium access protocol [219]. Another problem for TCP is the frequent route breaks and fluctuating routes. Even a short break in the path may force TCP to back off so much that it reduces the rate to the minimum. When the path comes back up again, TCP goes into the slow start phase, which means that the full bandwidth is not used for some time.

However, the expected real end-to-end performance still remains an open question. More research is required, especially measurements in real multi-hop networks, such as Clusters.

6.6 The Prototype

In this section, we outline the design and implementation of the modifications we did to enable LQA and high quality unicast routing within a Cluster in our prototype. We used a modified version of the Madwifi driver. Further, modifications were done to the Cluster formation application (ppand) and the routing daemon (olsrd). Otherwise, the prototype architecture introduced in Chapter 4 remained unchanged.

6.6.1 Madwifi driver

The Madwifi [129] driver is an open source driver for wireless cards based on the Atheros chip for Linux. This driver was chosen because its entire code is available, and because it implements many features usually implemented in firmware and running on the wireless card itself. This meant that we had access to the rate adaptation and link layer retransmission functionalities. For our experiments, we used Madwifi version 0.9.3.2 in combination with the 3Com OfficeConnect Wireless 108Mb 11g PC Card [1].

For some of the LQA proposals in this chapter, we needed access to extended information not usually required from a wireless card driver. In some cases, we could rely on the Linux standard interface for wireless drivers. The signal strength per packet and per neighbor is available through the IWSPY interface, which the Madwifi driver implements. For feedback of the retransmissions, however, no standard interface could be used. Instead, we implemented our own interface using the `/proc`-filesystem. Through a new file, we could export the number of successful transmissions and the number of retransmission attempts for each neighbor from the driver to user space applications, such as ppand. Those numbers were extracted from the rate adaptation part of the Madwifi driver, which needs that data for transmission rate selection. The ppand application periodically polls the file and by calculating the differences between consecutive polls, it can calculate the packet delivery ratio.

6.6.2 Modifications to ppand

The LQA implementation was done inside the link layer adaptation layer (LLAL) for WLAN in ppand, which we first introduced in Chapter 4. In addition to the hello packets, the LQA code periodically reads the received signal strength and the feedback from the retransmission mechanism. All this raw data is fed to a function that calculates the LQA, which is then sent via an IPC mechanism towards the routing daemon. The different LQA approaches tested were implemented as different functions, which could be selected at the compilation time. At the same time, all raw link quality data was dumped to a log file. Using that log file, the LQA based on other methods could be calculated afterwards for off-line testing and analysis.

6.6.3 Modifications to the routing daemon

All our experiments were carried out using a modified version of Olsr.org's OLSR daemon version 0.4.10 [169], which we simply call olsrd. Olsrd implements OLSRv1 [37] plus some extra options such as a link quality extension based on ETX using hello packets. To use the LQA values from ppand, we enabled this link quality extension, but replaced it with our own functionality that retrieves the LQA values from the ppand. For simplicity, olsrd still sent and handled its own hello messages (link quality-enabled hello messages) as usual. However, the reception (or the non-reception) of the hello messages was only used to determine the neighbor set, but not the link quality. A neighbor was excluded after ten consecutive hello messages were lost. The link quality information came from ppand via the IPC mechanism as explained earlier.

The ppand sent LQA updates for each of its neighboring Personal Nodes to the routing daemon with a frequency of 10 times per second. For obvious reasons, that was not the frequency used by olsrd to further disseminate the quality information. Instead, olsrd sent link quality updates, through OLSR's topology control messages, once every second for each link and each direction that the OLSR protocol decided to include in the partial link state set. This is a high frequency for OLSR (the standard proposes once per 5 seconds [40]), but was used in our small test bed for testing purposes. Future implementations should use a lower frequency and send additional updates whenever a significant quality change occurs.

Modifications to the link quality message formats were not required, since olsrd already provides message modifications for carrying link quality information. In olsrd, this information was included in the hello messages and the topology control messages using two 8 bits fields, which was enough also for our purposes. The only thing we modified, in addition to the retrieval of the LQA values from ppand, was the extra logging facilities. This was done for debugging and measurement purposes.

6.6.4 Lessons learned

The modifications to the Madwifi driver and olsrd were kept to a minimum and measured only 240 lines of code (LoC) and 190 LoC each. The ppand, where most of the LQA logic was located, measured 1650 LoC. However, that number includes all tested LQA variants in this chapter and repetitions of the same code from the wired LLAL implementation.

The CPU load of our test bed laptops remained at a moderate 10 % during all experiments. Most of the computation concerned the extended logging facilities and the generation and measurement of test traffic.

The LQA functionality was straightforward to implement as soon as we had determined which method to use. Finding the parts to modify and implement the required modifications of both the Madwifi driver and olsrd was relatively easy. The main difficulty was the amount of time required to get used to the code that needed to be modified.

The measurements themselves took a significant amount of effort. We had to try out different approaches and configurations in order to first find out what works and what does not. Then, proper scenarios that could exercise certain issues or trigger certain situations needed to be defined. Also this step could require several attempts. Finally, the actual experiment and the data collection could be carried out, before being compiled and presented.

One of our main challenges in the experiments was to assure that mobility experiments were exactly repeated and that fair comparisons could be done between two different experiments. It was not always possible to test two configurations in the same experiment. Instead, two different experiment trials were often required and each trial is unique. We tried our best to repeat the exact movements and speed and always averaged over 10 repetitions. However, this was done for only one mobile station. With multiple simultaneous mobile nodes, this problem will become more complex.

6.7 Summary

In this chapter, we took a close look at unicast routing within Clusters. There are plenty of unicast routing protocols for networks, such as Clusters, but they only provide a part of the solution. To make sure that the best possible path is selected, the routing protocol needs the best possible information of the quality of the potential links. This problem, we refer to as link quality assessment (LQA) and that was the focus of this chapter.

We investigated several approaches to LQA using available cross-layer information. Collected information came from the hello protocol as well as the data traffic and included data such as delivery ratio and received signal strength. We investigated approaches to combine these information sources in the best possible way to predict the data packet delivery ratio on a link. A few approaches were tested within our PN prototype in a three Nodes test bed. By using more cross-layer information, we could show that better paths were selected.

Further, we discussed general approaches to areas, such as asymmetric link qualities, heterogeneous networks, comparison between links with and without data traffic, and the effect of packet size. The quality of TCP streams over Cluster paths was also discussed.

Finally, we introduced how the prototype was extended with these LQA methods. We used the Madwifi driver, which is an open source driver for Atheros wireless chip-based cards, such as our 3Com OfficeConnect Wireless 108Mbps 11g XJACK PCCards. From the Madwifi driver, we could extract information that is not normally available to the operating system, such as feedback from the data packet retransmissions and acknowledgments. Then, ppand was extended to make use of this and other cross-layer information and perform the LQA calculations. The LQA results were then fed to the OLSR routing daemon, which used them for finding the best routes.

Chapter 7

PN Organization

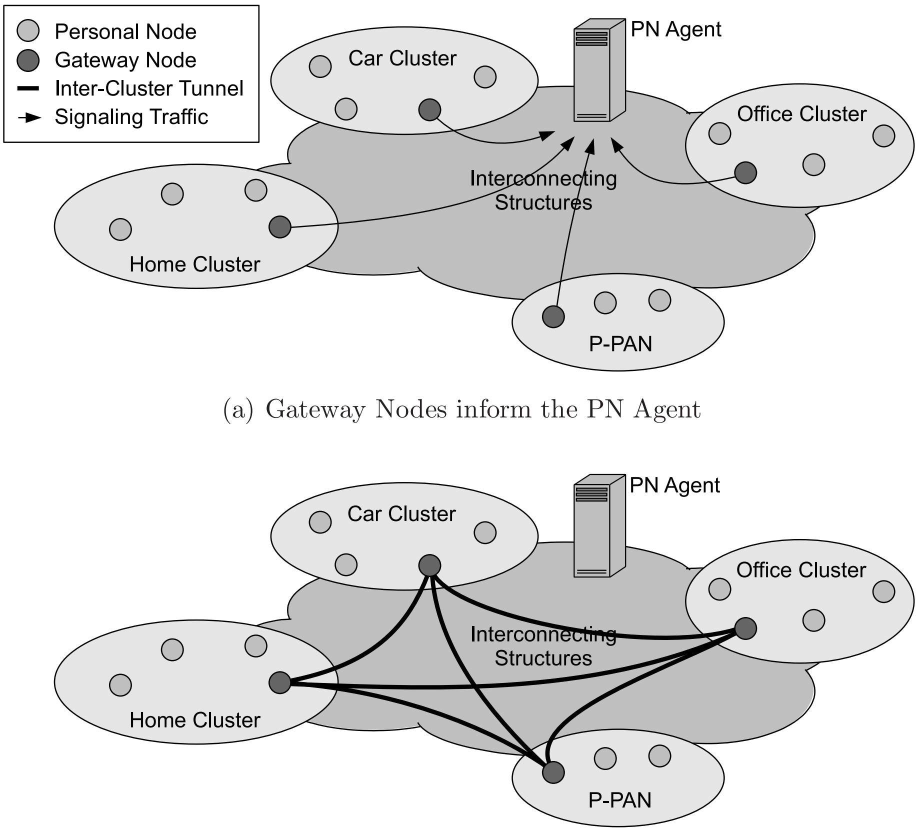
It is now time to move away from Clusters and personal local communication and look at the personal global communication picture. How can we connect remote Clusters to each other and how do we use the existing communication infrastructure to do that? We call this inter-Cluster communication and that is the topic of this chapter.

As we have explained earlier, a PN typically has multiple Clusters that are geographically dispersed. To form a PN and realize communication between the Clusters, three requirements need to be fulfilled [96]:

1. The Clusters need to have access to a fixed Interconnecting Structure through one or multiple Gateway Nodes.
2. Once access to the Interconnecting Structure is available, the Clusters need to be capable of locating each other in order to establish tunnels between them and thereby form a complete PN.
3. Finally, the Clusters must be able to maintain the tunnels regardless of Cluster or Node mobility.

The discovery and selection of Gateway Nodes that can provide access to the fixed infrastructure, is a Cluster internal issue and is best fulfilled through a Cluster-internal mechanism. For the second requirement to be fulfilled, we introduce the concept of a PN Agent. Clusters that have obtained access to the Interconnecting Structure announce their presence to the PN Agent as shown in Figure 7.1(a). The announcements contain information such as security credentials, current care-of addresses (CoAs) for the Gateway Nodes, and perhaps also the list of Nodes in the Cluster. The PN Agent can communicate this information to other Clusters and their Gateway Nodes, which may trigger the creation of secure tunnels between the Clusters as shown in Figure 7.1(b).

The purpose of the tunnels is twofold. First, they provide secure inter-Cluster communication by shielding the intra-PN communication from the



(a) Gateway Nodes inform the PN Agent

(b) Inter-Cluster tunnels are established

Figure 7.1: Example of PN establishment

outside world. Second, these tunnels are dynamically updated in order to handle the mobility of Clusters.

To handle the mobility, information regarding the availability of Gateway Nodes is propagated and the Gateway Nodes will react by dynamically updating the tunnels. When a Gateway Node changes its attachment point to an Interconnecting Structure, existing tunnels are destroyed and new ones are created. In cooperation with PN-wide routing and addressing, this results in a self-organized PN that consists of several Clusters interconnected by dynamic tunnels. This provides security and hides the Cluster mobility and Gateway Node changes from the Nodes in the PN.

This chapter is organized as follows. Section 7.1 introduces important requirements for inter-Cluster communication. Section 7.2 discusses related work. In Section 7.3, we cover PN-wide addressing. Section 7.4 discusses the aspects of infrastructure support. Section 7.5 covers the main functionalities of inter-Cluster tunneling, including mobility handling, tunneling strategies, Gateway Node coordination, and security. In Section 7.6, we continue with the PN routing aspects, including PN-wide broadcasting and QoS. Finally, in Section 7.7, we describe the implementation of the inter-Cluster communication extension to our prototype, before we summarize in Section 7.8.

7.1 Inter-Cluster Tunneling Requirements

For a successful implementation of inter-Cluster communication, it is crucial that the following requirements are met:

1. As much of the communication setup as possible must be automatic. It is unacceptable to require the user to manually bring connections up and down. This must be done automatically and intelligently by the system. However, some user control should be possible, such as permitting the use of access networks that are charged. The user may specify rules by which decisions are automatically made by the system.

2. The inter-Cluster communication must support mobility. This includes horizontal and vertical handovers between access networks as well as handovers between Gateway Nodes, whenever better access networks arise or used ones disappear.

3. As many existing access networks and Interconnecting Structures as possible must be supported. Emerging and future technologies, as long as they can be anticipated, should be supported. This means that current deployment setups must be supported, such as hotspots and digital subscriber lines (DSL) using IPv4, Dynamic Host Configuration Protocol (DHCP) [54], and network address translators (NATs). However, as future deployments evolve (e.g., using IPv6), they must also be supported. Hence, a great deal of flexibility is required.

4. It must not be necessary to change or add functionality to the current Interconnecting Structures. The deployment of PNs must not depend on certain technologies first being deployed in the infrastructure. This is not to say that additional functionalities that can improve the operation of PNs are not wanted, just that such functionalities must not be required.

5. The user's privacy must be retained, since a PN will carry user-related data. Communication going over various access networks and Interconnecting Structures must be properly protected offering both confidentiality and data integrity.

6. It is also important that inter-Cluster communication can achieve a high QoS whenever required. This implies that the PN must be able to choose the best Gateway Node, access network type, and access point for the communication. The quality of every option should, to some reasonable degree, be monitored or predicted so that a good selection

can be made. The selection may also be based on the needs of the applications active at every moment.

7. The inter-Cluster communication mechanisms should not waste energy and other resources. Unnecessary communication paths that consume energy or resources should be disconnected when not needed. This requirement may be in contradiction with the previous requirement on QoS, so a proper trade-off must be made.

7.2 Related Work

In this section, we list work related to inter-Cluster tunneling, including topics such as network mobility, secure tunneling, NAT-traversal, and packet relaying.

The most important mobility solutions for IP networks are IP Mobility Support for IPv4 (Mobile IPv4) [178] and IP Mobility Support for IPv6 (Mobile IPv6) [103]. They both require every mobile host to have a so-called home agent (HA) on their home network. Any host that wants to communicate with the mobile host sends packets to the home address, which is the address of the mobile host when it is on the home network. The HA intercepts those messages and forwards them in a tunnel to the mobile host, either directly or via a foreign agent (FA). The mobile host updates the HA when it's care-of address (CoA) changes, by sending a message with the new address. However, packets sent by the mobile host in the other direction are sent directly and not via the HA. This results in triangular routing, involving the corresponding host, the HA and the mobile host. However, in Mobile IPv6, it is also possible to send the CoA to the corresponding host so that all traffic can flow directly between the two hosts, thus avoiding triangular routing.

Mobile IP concerns mobility of single hosts, such as laptops. Hence, each node has its own mobility mechanism. If applied on a Cluster, which is a network of mobile hosts, each node must have its own mobility mechanism. When a Cluster roams together, all nodes need to update their HAs concurrently when the network changes point of attachment. For this reason, the Network Mobility (NEMO) Basic Support Protocol [49] was introduced. In NEMO, each mobile network has a mobile router (MR). The MR is connected to the infrastructure and has a home agent just as in Mobile IPv6 (NEMO is only specified for IPv6). Instead of a home address coupled to the home location, the MR has a home network prefix coupled to the home network. All the nodes in the mobile network get addresses with that network prefix from the MR via link local IPv6 router advertisements [158]. The nodes in the mobile network do not need to be aware of their mobility when the network roams, only the MR needs to take actions. In NEMO basic support, all traffic to and from the mobile network is tunneled via the home agent. However, route optimization has been proposed [161].

NEMO has several limitations and drawbacks. First, the mobile network needs to be a single-hop network since it relies on link local router advertisements. Hence, NEMO cannot support multi-hop Clusters without additional functionality. One option would be to use Ananas [33], which makes a multi-hop network behave like a normal single-hop LAN for the IP layer by introducing an intermediate layer. As described in Section 4.2, Ananas is not ideal and hence a better solution is to replace the network prefix advertisements with a solution that works over a multi-hop network. Another problem, not fully solved in neither NEMO nor Mobile IPv6, is multi-homing. Currently, IETF is studying solutions for this problem [149] [160], including soft handovers between two access networks as well as multi-homing for increased reliability and load sharing. In NEMO, there is the additional problem of multiple MRs in a single mobile network. If the two MRs have different network prefixes, then it becomes difficult for mobile nodes to switch from one MR to the other. Hence, when using NEMO for PNs, all the MRs belonging to the PN need to have the same prefix. Unfortunately, this is difficult to achieve as the HA would not know which MR to forward a packet to when the MRs are no longer in the same mobile network (Cluster). Because of these limitations in NEMO, we do not propose to use NEMO for inter-Cluster communication.

The work in IETF on mobile nodes and networks is currently quite intense. An important area being in focus is that of detecting network access (DNA) [36] [157]. Whenever a mobile node discovers access to a new network on one of its interfaces, it needs to detect the IP configuration of that network as well. Unfortunately, the IPv6 neighbor discovery protocol (or the DHCP for IPv4) is not very fast and this is what DNA tries to remedy. It is also worth mentioning that timely support from the link layer, similar to our work in Chapter 6, is crucial for achieving smooth handovers. We will discuss this topic in Section 7.6.2. Nevertheless, as soon as DNA is becoming available, Gateway Nodes should be made aware of it and start using it when connecting to the Interconnecting Structures.

Another potential candidate for inter-Cluster communication is the Host Identity Protocol (HIP) [151]. In HIP, each host has at least one fixed identifier called host identifier (HI). The HI may have the same form as an IP address and is used when establishing TCP or UDP sessions. The HI is mapped to a current valid and topologically correct IP address of the host. To initiate a new HIP session, a corresponding host would need to know the host's HI and its current IP address. This can be discovered if they are direct neighbors on the same network. If not, the host may have a rendezvous server (RVS) [116], which is published together with the HI in, for instance, DNS. Each host will keep its RVS up-to-date by informing it through HIP messages every time it changes address [115]. When a HIP session is established with another host, a change of the IP address is also communicated to the other host by a HIP message [165]. This will speed up the mobility handling for the ongoing session. Extensions for NAT-traversal also exist [111].

The data traffic of HIP is typically tunneled using IPSec [110] with ESP [109][105], but other alternatives are possible. Furthermore, this is not the only security feature of HIP. For instance, the HI is actually the public part of an asymmetric key. Using this feature, the end hosts can be authenticated and all HIP messages and data traffic can be protected after a four-way handshake between the two hosts. Unfortunately for PNs, HIP has its own security mechanism, which significantly differs from the one proposed earlier for PNs. Hence, for use in PNs, the HIP security mechanisms must be replaced by the pair-wise keys approach used in a PN. Furthermore, HIP does only do host-to-host communication and not network-to-network communication. Hence, for PN inter-Cluster communication, HIP must be extended with routing over the HIP sessions, which need to become HIP tunnels. However, several other HIP mechanisms can be utilized in one way or another, such as the mobility handling, the RVS mechanisms, and the NAT-traversal.

Mobility can also be handled at the transport layer [142][207][12][238] or the application layer [203]. The principles are the same as for the various IP-level mobility solutions. To maintain ongoing sessions, either a proxy is inserted between the two peers as in MSOCKS [142], or redirection messages are sent between the peers as in Migrate [207], STEM [12], and Reliable Sockets/Packets [238]. The main difference lies in the implementation, whether both TCP and UDP are supported, and whether modifications are needed to the applications and/or the operating system. To locate a mobile node and discover its current IP address, most protocols suggest using the DNS UPDATE protocol [227]. However, when using SIP [203], it can be used for both maintaining ongoing sessions during mobility and discovering the current location of mobile nodes. As with the network layer mobility support protocols, any of these protocols could be a part of the solution for the inter-Cluster communication framework. However, we do not really propose to use any of them because the lack of a clear advantage in doing so or the very specific implementations provided by some of the proposals.

MOPED [114], which was introduced in Section 2.2.9, provides a more complete alternative for inter-Cluster communication. A person's personal devices, which can connect directly with each other, form so-called components. Components are essentially the same as Clusters. A proxy server located somewhere in the infrastructure keeps track of all the personal devices, components, and how they are connected. The proxy knows where each device is and can therefore solve issues such as addressing, routing, load balancing, and mobility. The mobility between the proxy and the perimeter (Gateway Node in PN terms) of the components is done through low overhead tunnels whose mobility is handled by Mobile IP. A MOPED forms a star-topology with the components in the edges and the proxy in the center, which means that all inter-component traffic goes through the proxy. Route optimizations are discussed, but only for traffic going in or out of a MOPED. Furthermore, no security, privacy, or anonymity solutions are discussed. Otherwise, MOPED provides most of the functionalities of a PN and in a similar way.

Peer-to-peer techniques such as Chord [214] and Pastry [196] may also provide a good alternative for inter-Cluster communication. One solution built on peer-to-peer techniques is Robust Overlay Architecture for Mobility (ROAM) [244], which is based on Internet Indirection Infrastructure (i3) [213]. In ROAM, servers in the infrastructure relay packets to the correct destination using special tags. Each host has a unique tag that is associated with one of the servers. Mobility is supported when the hosts update their destination addresses associated with their tags at the ROAM/i3 servers and this could be used for handling inter-Cluster mobility. However, the non-direct routing still remains a question.

7.3 PN Addressing

Before tackling PN-wide inter-Cluster communication, we first need to decide on the internal addressing of Personal Nodes. Each Node should have a fixed intra-PN address that stays the same as long as the Node is part of the PN. Since we assume that Personal Nodes can roam freely, there is no possibility for a hierarchical organization of intra-PN addresses without introducing address changes. If the intra-PN addresses still change, then the mobility problem is not entirely solved. Hence, the intra-PN address should remain fixed.

The only remaining problem is the assignment of this address, which must happen as soon as a Node joins the PN and then should remain static during its entire membership to the PN. The address can be self-assigned, but must be unique within the PN. World wide unique identities or addresses, such as the EUI-64 [91], can be used as the intra-PN address if they are not too long and if the Node has one. However, as we explained in Chapter 3, we do not really want to use such globally unique addresses when there is no need for them. Furthermore, 64 bits (or 48 bits) Node addresses are unnecessary long for our purpose, if we assume that PNs might have no more than hundreds of Nodes. An 8 or 16 bit Node address plus prefix is therefore more suitable. A longer prefix will allow for better compression in routing and neighbor discovery packets based on the common MANET signaling packet format [38].

If we want a short Node address, the chance of address conflicts becomes significant. Hence, there is a need for duplicate address detection (DAD) [30]. The first step a new Node should take, after becoming a member of the PN and assigning itself an intra-PN address, is to verify the uniqueness of its new intra-PN address with the Node it was paired with. A Personal Node will have Trust Relationships with most, if not all, Personal Nodes in the PN. Hence, it will know most of the already used addresses in the PN. If no address conflict is found, it is most likely unique within the PN, but further DAD is still advisable. Here, the PN Agent could be queried for absolute certainty. The PN Agent should know about every Node in the PN. If a Personal Node is unknown to the PN Agent, it is practically not yet part of the PN. Another option is to rely on one of the DAD schemes for MANETs [30]. However, they all are much more complex and create much more overhead compared to just querying the PN Agent as explained above.

7.4 Infrastructure Support

We obviously need the support of Interconnecting Structures to connect the Clusters. The only question is how much more support is required compared to what is currently offered [186]. IP transport between Gateway Nodes that are connected to the Interconnecting Structures is essential, but support from special servers may also benefit the operation of a PN. A contactable server somewhere in the Interconnecting Structures can offer services similar to a home agent in Mobile IP or a rendezvous server in HIP. We refer to such servers as PN Agents. Furthermore, the burden of Gateway Nodes may be reduced by special functionality offered by the access router that the Gateway Node connects to as been suggested by the MAGNET project [133]. Such routers with special PN functionality, we refer to as Edge Routers.

7.4.1 PN Agent

The PN Agent is a management entity located anywhere in the Interconnecting Structure or elsewhere from where it can be reached all the time (e.g., on a non-mobile Gateway Node). Each PN has a PN Agent and its task is to keep track of all Personal Nodes and Clusters in a PN. All the Gateway Nodes need to be aware of the IP address of their PN Agent. Therefore, the address is distributed to every Personal Node during the personalization. Gateway Nodes, which are always also Personal Nodes, hence know the address of the PN Agent.

Clusters that have obtained access to the Interconnecting Structure announce their presence to the PN Agent as shown in Figure 7.1. More precisely, the Gateway Node sends a registration message to the PN Agent. The information contained in the registration messages must be transferred in a secure way so that the information in the messages cannot be altered and is invisible to non-authorized parties. The registration messages need to contain at least the following essential information: PN identification, Node identification (e.g., intra-PN address), and the care-of addresses (CoA) of all it's active attachment points to the Interconnecting Structure. The PN identification is needed since there might be more than one PN Agent running on one server. The PN Agent must also be able to check the credentials of the Gateway Node to access a certain PN and the message's authenticity. The Gateway Node's CoAs are of course needed, as this will represent the endpoints of the inter-Cluster tunnels. The PN Agent stores this information from all the Gateway Nodes in a secure database.

The information stored in the PN Agent can be queried by the Gateway Nodes in order to establish the inter-Cluster tunnels. Alternatively, the PN Agent may decide which tunnels should be established or maintained. The PN Agent can also assist in establishing tunnels between two Gateway Nodes that cannot directly establish a tunnel due to NATs and, if even that is impossible, the two Gateway Nodes may send their data traffic via the PN Agent.

The purpose of the PN Agent is quite similar to the home agent in Mobile IP or the rendezvous server in HIP. In this light, the PN Agent is best seen as an abstract entity and we could base the Gateway Node to PN Agent protocol on either Mobile IP or HIP. Mobile IP might not be the best choice, due to the difficulty of achieving direct tunnels between Gateway Nodes without modifying the protocol. HIP, on the other hand, implements functionalities that we do not need. As will be shown later, we will propose a protocol based on a simplified version of HIP, where the security parts of HIP are replaced with the PN Trust Relationships.

Note that the PN Agent also can provide additional functionality. It may assist in other PN-internal mechanisms such as name resolution and service discovery. Further, the PN Agent can be used by Foreign Nodes that wish to communicate with the PN. In that case, the address of the PN Agent is the only address a Foreign Node needs to know in order to be able to communicate with the PN. This will be discussed further in Chapter 8, where we will address the communication between PNs and Foreign Nodes.

Even though we have described the PN Agent as a single node in the Interconnecting Structure, it is not necessarily so. The PN Agent could be a distributed functionality running on several servers, a set of redundant servers [226], or a peer-to-peer network of servers (such as Chord [214]). There can be many reasons for for a distributed PN Agent, but the two most important reasons are increasing availability of the PN Agent functionality and reducing response time experienced by the Gateway Nodes.

Another important aspect is where the PN Agent server or servers should reside and under whose control and responsibility they are. The PN Agent can be a server in the Interconnecting Structure operated by the PN owner himself. A user that needs or wants total control of the PN Agent may want to run the PN Agent functionality on one of his own nodes, such as a Gateway Node in the home Cluster that connects to the Interconnecting Structures using a reliable and fixed connection without NAT. Alternatively,

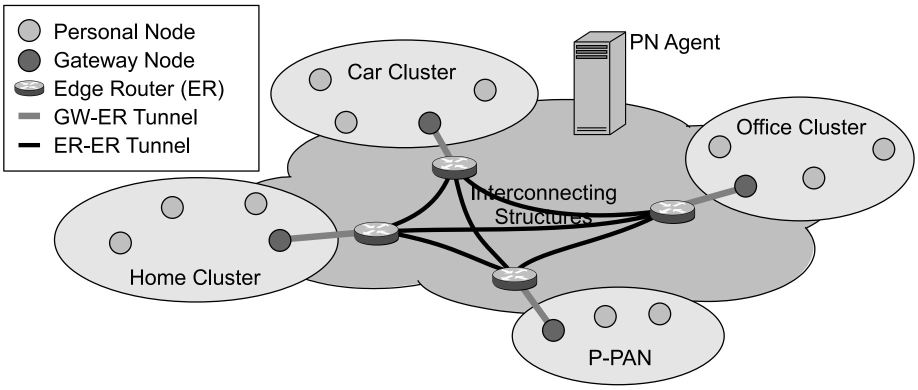


Figure 7.2: Inter-Cluster communication with Edge Routers

network or service providers may offer PN Agent functionalities that can be used by their customers. We hope it is clear that there are several options for PN Agent deployment and which option is the best is not only a technical matter, but has, for instance, also business consequences.

7.4.2 Edge Routers

An Edge Router (ER) is an access router that sits on the edge of the Interconnecting Structure, communicates with the Gateway Nodes and supports them by offering special functionality for PNs [136][125][76]. They need to be managed by a network or service provider and thus will probably be owned by the provider. On behalf of a Cluster, an ER can perform several intra-PN tasks, such as communicating with the PN Agent and taking care of the inter-Cluster tunnel establishment and management. In this way, ERs can relieve the Gateway Nodes of some of the work and thereby allow them to reduce their power consumption and resource requirements. Figure 7.2 shows an example of inter-Cluster communication using ERs. In this example, we assume that not all access networks provide ER-functionality and hence some Gateway Nodes still need to perform all tasks related to inter-Cluster tunneling.

If we assume that Gateway Nodes need to maintain many tunnels, then this maintenance consumes valuable resources, such as processing and battery power. The tunnel maintenance may therefore overload the Gateway Nodes, which are often mobile and battery-powered. Thus, it will be useful to let ERs, which are fixed and powerful devices in the Interconnecting Structures, support the tunnel establishment as much as possible and thereby place the overhead needed for establishing and operating a PN in the Interconnecting Structures. Furthermore, ERs can assume other responsibilities as well, such as inter-Cluster routing, remote service discovery, service repository, and more.

The use of ERs has both advantages and disadvantages [186]. Let us summarize the advantages of an ER-based solution as follows:

1. Some tasks can be carried out by the ER, which leads to less consumption of scarce resources in mobile devices.

2. Mobile Gateway Nodes can be made more lightweight. This leads to simpler devices that have lower cost and less power consumption. Consequently, more PN Nodes can provide Gateway Node functionality, which leads to increased flexibility in accessing the Interconnecting Structures.

3. PN formation and maintenance are faster and hence can better support Cluster mobility.

4. ERs may support special functionality that can optimize handovers that current access technologies do not offer. This could, for instance, include Fast Handover for Mobile IPv6 (FMIPv6) [112] and/or Hierarchical Mobile IPv6 Mobility Management (HMIPv6) [208].

The ERs are infrastructure-based entities that contain functionality explicitly designed for PNs. However, this approach has serious drawbacks:

1. ERs need to be deployed. Since ERs are access routers uniquely designed for PNs, it is necessary modify the infrastructure by introducing these network elements. This has been proven to difficult in the past and therefore is likely to be a major stumbling block. If Gateway Nodes require ERs, then the deployment of PNs can only take place after operators have invested on a sufficient scale in ERs and hence there is a risk that this will slow down the success of PNs. Furthermore, the service providers need to maintain these more complex ERs.

2. ERs do not reduce the complexity of the PNs. Due to the expectation that there will be many access networks without ERs, it is still necessary for the Gateway Nodes to implement full Gateway Node-functionality. As long as not all access networks offer ER-functionality, this will be a drawback. In the mean time, Gateway Nodes need to handle two cases: access networks with ERs and access networks without ERs.

3. The ERs need to be trusted by the user since ERs will support the internal mechanisms of the PN and this may endanger the security and privacy of PNs. In the case that an ER is not trusted, the Gateway Node cannot use the ER and must instead perform all Gateway-functionalities itself. Furthermore, ERs must trust the PN Agents, even when they belong to another operator or to the user himself.

The issue of ER deployment is a major drawback that currently makes it important to support ER-less access networks. Future solutions may consider ER technologies as a way to optimize the performance of PNs. We will therefore mainly focus on solutions not requiring ERs in the remainder of this thesis.

7.4.3 PN networking without infrastructure support

It is of course possible to design an intra-PN communication system without special infrastructure support such as ERs and PN Agents. To do so, we need to turn our attention to peer-to-peer technology. The biggest advantage of peer-to-peer technology is that, in principle, it can make infrastructure-based support completely unnecessary. The peer-to-peer system ROAM [244], which we introduced in Section 7.2, unfortunately does not demonstrate this advantage. Instead, ROAM requires the deployment of i3 servers and it is these servers that form the peer-to-peer network.

To make both ERs and PN Agents superfluous, we need to make the Gateway Node themselves into peers in the peer-to-peer network. This means that the Gateway Nodes need to manage the inter-Cluster tunnels themselves. Each Gateway Node needs to remember the current locations (i.e., the CoA) of as many other Gateway Nodes in the PN as possible. This will work if some Gateway Nodes almost never move or if not all Gateway Nodes move at the same time. Normally, we can expect that at least one Gateway Node (e.g., the home-Cluster Gateway Node) never moves, which means that the other Gateway Nodes always can connect to that Gateway Node to be updated.

There are two main problems with not having the support of a PN Agent. The first major problem is the bootstrapping of the peer-to-peer overlay. In the beginning, all Personal Nodes need to gather in one single place and form one single Cluster in order to exchange the CoAs. New Gateway-capable Nodes added to the PN need to communicate with a connected and updated Gateway Node in order to retrieve all current CoAs. Whenever a Node has been deactivated for a long time, this procedure might need to be repeated. There is always the risk of some of the Gateway Nodes being disconnected from the rest and not knowing how to reconnect to the PN, even though it may work most of the time.

The second major problem in PNs without a PN Agent is the slow response to mobility. When a tunnel needs to be updated because an endpoint has moved, it is important that a new CoA or an alternative Gateway Node is found quickly. Unfortunately, when also several other Gateway Nodes have disappeared, it may take a while before the right Gateway Node has been queried. The alternative is to update all the other Gateway Nodes all the time, which, of course, introduces a lot of overhead. Because of these reasons, and the reason outlined in Chapter 8 regarding foreign communication, we propose to always make use of a PN Agent.

7.5 Inter-Cluster Tunneling

Inter-Cluster tunneling can be done in several ways. The main point is that intra-PN packets are encapsulated in encrypted IP packets that travel over the Interconnecting Structures between Gateway Nodes. In addition to this, the packet overhead should be minimum and the encryption lightweight. We propose to a solution similar to IPSec and UDP-encapsulated ESP [80], but without the security parameter index (SPI). Instead, we propose to use the pair-wise keys for derivation of session keys. This does not achieve the lowest possible overhead, but will serve as a good example.

A tunnel is a connection between two tunnel endpoints (TEPs). The two TEPs should belong to two different Personal Gateway Nodes in two different Clusters in order to be useful. A TEP is nothing more than a routable CoA and a port number belonging to an active interface on a Personal Gateway Node. Extra parameters can be associated with a TEP, such as NAT information and expected QoS. The TEP information is shared among the Gateway Nodes and the PN Agent so that tunnels can be established. In this section, we will discuss questions related to which tunnels to establish between which TEPs, how to communicate TEP updates efficiently, how to handle handovers and mobility, as well as NATs and security.

7.5.1 Mobility and dynamic tunneling

Every Gateway Node informs the PN Agent about all its active TEPs. If a Gateway Node has more than one valid IP address for a given active interface, it may announce all of them as TEPs, if it makes sense. Reasons for this may be that the different IP addresses use different paths through the Interconnecting Structures or that an access network supports both IPv4 and IPv6. Consequently, it is normal for a Gateway Node to announce several TEPs, especially if the Node is multi-homed. However, each TEP must be routable through the Interconnecting Structures so that it can be used by the other Clusters.

The PN Agent keeps a complete database of all active Gateway Nodes in the PN and their active TEPs. Any Gateway Node may query this database for information about the other Clusters, Gateway Nodes, and their valid TEPs. They may also subscribe to updates regarding other Clusters and Nodes so that they can receive updates in a timely fashion. In addition to the complete database kept at the PN Agent, each Gateway Node keeps a partial database over the TEPs and Clusters related to all its established tunnels. Figure 7.3 shows an example of how the TEP information is distributed over the PN Agent and the Gateway Nodes. If a tunneling approach that keeps all tunnels up is used, then all the local databases on the Gateway Nodes will be complete. In addition to the TEP information, each Gateway Node also keeps information related to the tunnels, including security keys, QoS,

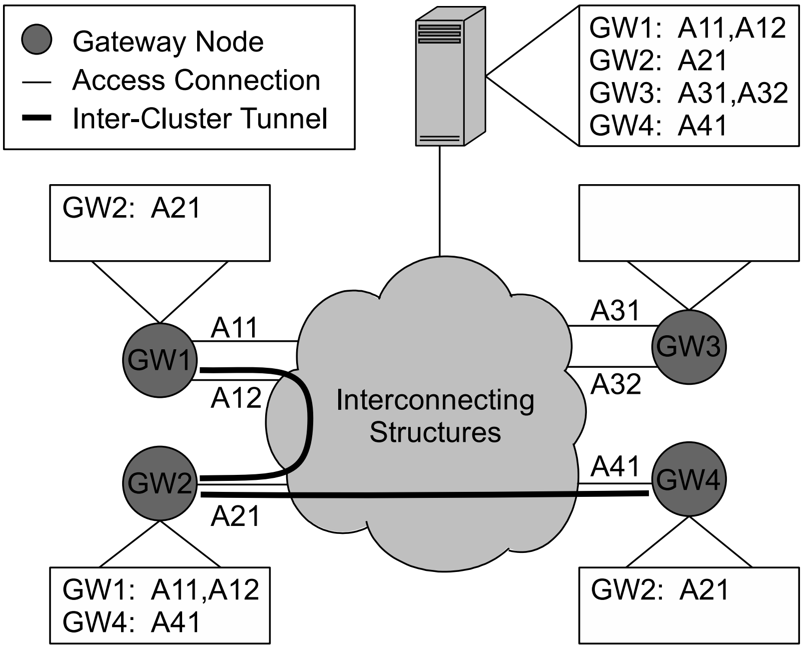


Figure 7.3: Inter-Cluster tunneling and the TEP databases

not yet delivered packets, etc.

The Gateway Nodes use the TEP information to establish tunnels among themselves. TEP information is exchanged periodically and when changes occur between the Gateway Nodes and the PN Agent. Since connections suddenly may disappear, all TEPs are soft states and need to be updated periodically. However, when possible, changes to the active TEPs must be announced timely to concerned parties within the PN. A Gateway Node that sees one of its TEPs disappearing or a new TEP appearing must always inform the PN Agent about the change. Other Gateway Nodes in the PN may automatically receive information through the PN Agent. To speed things up and to improve reliability, the Gateway Node may send the update information directly to the Gateway Nodes with which it has established tunnels. In this way, even less data packets will be affected due to the speedy routing of signaling packets. Figure 7.4 shows an example of how the updates are communicated when a Gateway Node (GW1) looses one of its TEPs and finds a substitute.

When a Gateway Node receives TEP updates from other Gateway Nodes or from the PN Agent, it may need to take some actions. It may need to disconnect an active tunnel and establish an alternative tunnel to an alternative TEP on the same Gateway Node or to another TEP on another Gateway Node in the same Cluster. If the TEP update contains new TEPs, it may want to switch to the new TEP, e.g., because of the expected QoS is better. All this is up to the tunneling strategy, which may be influenced by the user's preferences.

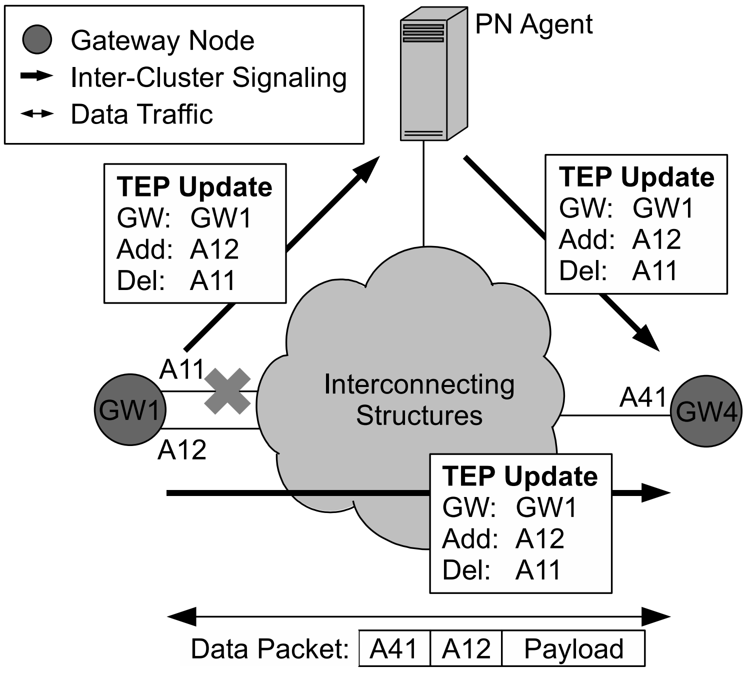


Figure 7.4: Inter-Cluster tunnel update due to mobility

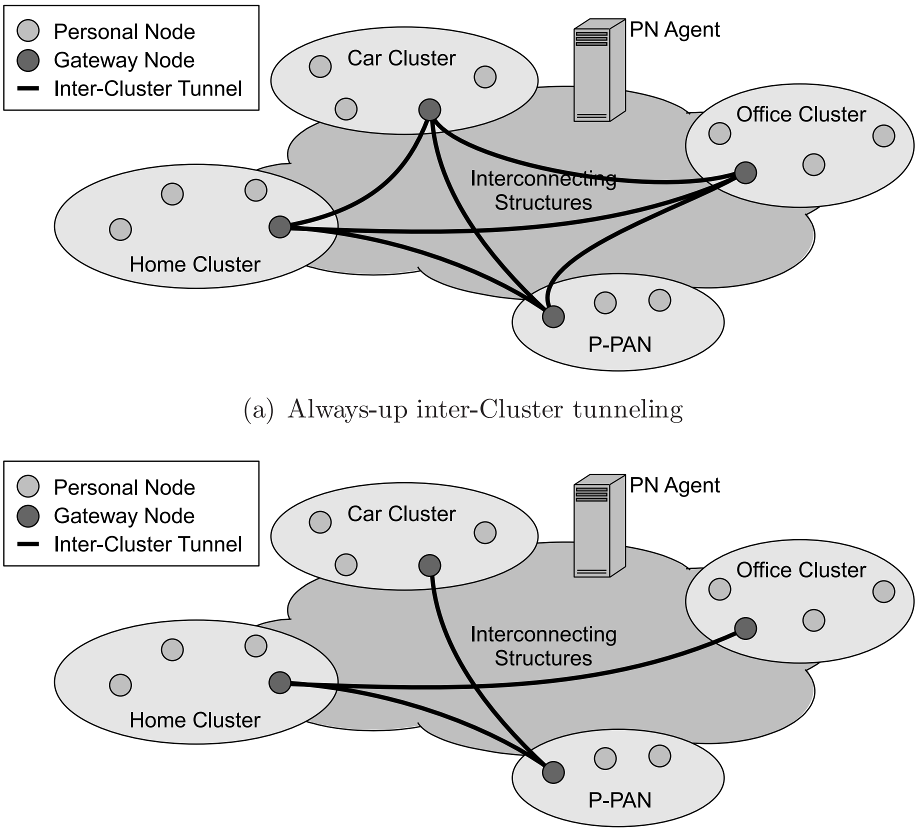
7.5.2 Always-up and on-demand tunneling

In an always-up tunneling approach, all possible inter-Cluster tunnels will be up and running at all times, even if there is no traffic. This is a sort of proactive approach.

There can be several tunnels between two Clusters or even between two Gateway Nodes. Multiple tunnels may be useful for redundancy, fault tolerance, and increased throughput. These arguments, in combination with the simplicity of the approach, favor an always-up tunnel maintenance policy in which tunnels are established and maintained as soon as a Cluster is connected to the Interconnecting Structure.

The Gateway Nodes initiate the tunnels with the help of the PN Agent, build a quasi-permanent connection with all present Gateway Nodes in the PN and keep these tunnels intact as long as possible. When the attachment point of a Gateway Node changes due to mobility or other reasons, it causes all the tunnels using the old TEP to be diverted to the new TEP. The idea here is to maintain the tunnels proactively between all Gateway Nodes so that there are valid tunnels between all Clusters at all times.

The alternative to always-up tunneling is on-demand tunneling; a sort of reactive approach. Figure 7.5 shows the difference between these two types of tunneling approaches. In the case of on-demand inter-Cluster tunneling, tunnels between Gateway Nodes are only established when needed. This means that a Gateway Node only sends TEP updates to the PN Agent and



(a) Always-up inter-Cluster tunneling

(b) On-demand inter-Cluster tunneling

Figure 7.5: Example of PN establishment

the Gateway Nodes with which it has active tunnels and the PN Agent only sends updates to Gateway Nodes that need them. Except for this, the two approaches work in the same way.

The difference between an established tunnel and a pair of TEPs without an established tunnel is actually not that big. With an established tunnel, the two Gateway Nodes need to make sure that their session keys (the security associations when using IPSec-based tunneling) are installed and being rekeyed and that TEP updates are timely and correctly communicated between the two Nodes. However, the biggest difference is whether the Gateway Nodes need to keep the access network of the TEP up and running, because there might be significant power savings in doing so. A Gateway Node that does not have any established tunnels may be able to disconnect all its access connections. However, one must not allow all Gateway Nodes to disconnect all their access networks since this will make the Cluster non-reachable from the other Clusters and the PN Agent.

On-demand tunneling may also establish multiple tunnels between two Clusters. This requires a smart mechanism that can decide whether a tunnel should be established or not. It needs to consider the needed power and overhead for maintaining each extra tunnel and weigh that against the required QoS and resilience. Hence, on-demand tunneling needs to decide whether there should be none, one, or more tunnels between two Clusters and this makes the approach more complex than the always-up tunneling approach.

7.5.3 Gateway Node coordination

When there are several Gateway Nodes with several available overlapping or non-overlapping connections to the Interconnecting Structures, there might be a need for coordination among the Gateway Nodes. If two Gateway Nodes in the same Cluster have access to the same access point, there is little benefit of keeping both active at the same time. Keeping a connection active costs energy that can be saved if one of the connections is terminated. Also when the used bandwidth in or out of the Cluster is small, connections can be terminated so that no idle connections are unnecessarily maintained. However, when existing connections drop out or the bandwidth needs increase, these terminated connections should be brought up again. Hence, what is needed is a mechanism that can bring up and down TEPs on the Gateway Nodes in the Cluster. Bringing down a TEP can be anything from only stop announcing its existence so that it is never used to also stop maintaining the connection or completely shutting down the entire interface. It should also be noted that connections and interfaces may periodically be brought up again to discover new connection possibilities and to determine whether the old ones still exist.

To achieve this, a Gateway Node coordination protocol that operates within a Cluster is needed. Information about available connections to the Interconnecting Structures, whether they are up or down, as well as their current QoS and current load should be shared among the Gateway Nodes. Decisions can then be taken and again communicated back to the Gateway Nodes for readjustments. The protocol must understand the cost of keeping an interface up and running, cost of maintaining a link, achievable QoS, as well as user preferences. While the exact details of such a protocol still need to be worked out, it is clear that this protocol needs many-to-many communication among the Gateway Nodes. For this, we propose to use the flooding protocol we proposed and investigated in Chapter 5.

7.5.4 NAT traversal

One of the main challenges of inter-Cluster communication without any additional support from the Interconnecting Structures is the handling of network address translators (NATs). Most access points, digital subscriber lines (DSL), or other types of Internet access that are currently offered to end customers provide IPv4 with NAT. Consequently, to be useful today, inter-Cluster tunneling must possess the capability to traverse NATs.

When the Gateway Nodes are not on publicly routable IP, the PN Agent can assist in establishing the tunnels so that they can traverse the NATs. If only one of the TEP for a new tunnel is behind a NAT, a message sent via the PN Agent can trigger the Gateway Node, with the TEP behind a NAT, to initiate the tunnel. If both TEPs are behind NAT, a method such as simple traversal of UDP through NATs (STUN) [194] can be used. In the case that the NATs do not allow such mechanisms (i.e., they are symmetric/restrictive) or there are additional firewalls involved, relaying of tunnel packets is an option. Traversal using relay NAT (TURN) [195] is a well known example of such an approach to NAT traversal.

If many Gateway Nodes need assistance with tunneling, the PN Agent may need to be powerful and have a good network connection. An alternative option is to delegate this to other known relay servers not behind NATs. Such relaying servers could be other Gateway Nodes in the PN that are not behind NATs, Gateway Nodes from other PNs, or dedicated relay servers available in the Interconnecting Structure. These servers could be organized and discovered through peer-to-peer networks similar to the way Chord [214], Pastry [196], or Skype [206][67] operates. If there are plenty of such relay servers around, then the effect of triangular routing can be minimized when servers nearer the two Gateway Nodes are selected.

7.5.5 Tunneling and signaling security

It is obvious that data traffic that crosses an external Interconnecting Structure needs to be protected. This includes encryption of the entire data packets, integrity protection against unauthorized alterations of the data packets, and mechanisms against replay and DoS attacks. When two Gateway Nodes establish at least one tunnel between themselves, they negotiate session keys. This negotiation is protected using the already deployed pair-wise key that they both have and share. The session key should not be associated with any particular TEP or CoA. Instead, they should be associated with the Gateway Nodes' PN-internal addresses. This enables the establishment of a new tunnel using different TEPs without first negotiating new session keys and this is important for timely handovers when TEPs disappear. Each packet needs to contain the encapsulated packet with the PN-internal IP header, including information such as the PN-internal addresses of the destination and source Gateway Nodes as well as a packet counter. The additional information is required for successful protection against replay attacks and requires both Gateway Nodes to keep a short list of already received packets. It should be noted that nothing of this is entirely new since the mechanisms used by IPSec in HIP [105] are almost identical.

Not only data traffic needs protection, also signaling traffic needs protection. Signaling between two Gateway Nodes can be protected with the same mechanisms as for the data traffic. However, signaling with the PN Agent must also be protected. One option is to install pair-wise keys between the Personal Nodes and PN Agent as if the PN Agent also was a Personal Node. The pair-wise key can be used in the same way as above to protect the Gateway Node to PN Agent signaling traffic.

Regarding anonymity, we need to trust the access network providers in this case. The encryption of PN and Node identifiers is not sufficient for good anonymity protection since the destination addresses may reveal the identity of the user anyway. Packets sent to the PN Agent address and a semi-permanent Gateway Node CoA, such as the home network Gateway Node, should be enough to deduce the identity of the user. Hence, PN and Node identifiers can better be sent unencrypted for performance reasons. A better approach to retain privacy is to only use trustable network providers and make sure packets between the Gateway Node and the network provider's access point is properly protected and encrypted. In this way, only the network operators have access to this kind of information, which should be enough for most users.

A Gateway Node can easily connect to an access network without revealing its identity to any other than the network provider. The access point (or base station) advertises itself in the clear. Assuming that the PN received a public key of that access network provider when the user of the PN signed up for the service, the Gateway Node can establish a secure and authenticated channel using that key. After this, the access point can verify the identity of the Gateway Node without any adversaries being able to eavesdrop. This scheme works because the access point has no reason of being anonymous and hence advertises its identity in the clear, which means that the Gateway Node knows exactly which key to use.

7.6 Inter-Cluster Routing

When using an always-up tunneling approach, standard routing protocols becomes an option. An ad hoc routing protocol, such as DYMO [32] or OLSR [40], is advisable due to the mobile nature of PNs and the flat addressing structure. However, when using on-demand tunneling, the routing protocol must be capable of bringing up inter-Cluster tunnels when needed. In either case, the special topology of a PN, which consists of Clusters with tunnels between them, asks for a more tailored approach. For instance, there is very little benefit of sending full topology information between the Clusters. This will only add overhead to the perhaps very limited access connections. Instead, we propose a scheme that communicates minimal information, but still is able to achieve good routes. It involves the PN Agent, is suitable for both the always-up and on-demand tunneling approaches, and is described in the following section.

7.6.1 PN Agent-based routing

The active TEPs in a Cluster are determined by the Gateway Node coordination protocol outlined in Section 7.5.3. However, whether to use these TEPs and establish tunnels must be determined by the routing substrate. To achieve this, a Gateway Node needs to know in which Cluster a certain Node is and, of course, which TEPs can be used to establish a tunnel to that Cluster. One good approach is to let the PN Agent also know the Cluster member Node list so that other Gateway Nodes can inquire about this as well.

Consequently, every Gateway Node informs the PN Agent about the member Nodes in its Cluster. The list can be retrieved from the intra-Cluster routing protocol if a table-driven protocol (e.g., OLSR) is used. If a reactive protocol is used, there might be a need for a special mechanism that discovers the Nodes in a Cluster. In either case, the PN Agent should constantly be updated so that changes are propagated to the rest of the PN when needed.

For each destination Node, the PN Agent knows which Gateway Nodes can be used and which active TEPs those Gateway Nodes have. When a Gateway Node needs to send a packet to a Personal Node outside its own Cluster, it sends an inquiry to the PN Agent. In response, it will get a list of Gateway Nodes and their TEPs that can be used. It selects the best TEP, establishes a tunnel from its own best TEP to that TEP, and then sends the packet across. At the same time, it subscribes to the PN Agent for updates related to the remote Node. That is, it will be informed of any update regarding any Gateway Node in the same Cluster as the Node so that proper actions can be taken when necessary. For instance, switch to a better TEP if one becomes available.

It is worth noting that the PN Agent does not really need to know exactly which Clusters exist. Since Clusters merge and split that would be an extra unnecessary burden to track. Ultimately, only a list of Gateway Nodes and their TEPs that can be used to communicate with a particular remote Personal Node are needed.

The next question concerns the integration between the PN Agent-based PN routing and the intra-Cluster routing protocol. Since there is no difference, except in QoS, whether packets are sent via one Gateway Node or another, it is up to the intra-Cluster routing protocol to decide which one to use. When using OLSR, we can use the attached network set functionality of OLSR to advertise the Gateway Nodes inside the Cluster. Using LQA of the links within the Cluster in combination with quality assessment of the available connections to the Interconnecting Structures should result in the best possible path at all times. In Figure 7.6, assuming that Node 1 wants to send a packet to Node 7, the intra-Cluster routing protocol of the left Cluster may choose Gateway Node 2 since it may conclude that path A is better than path B.

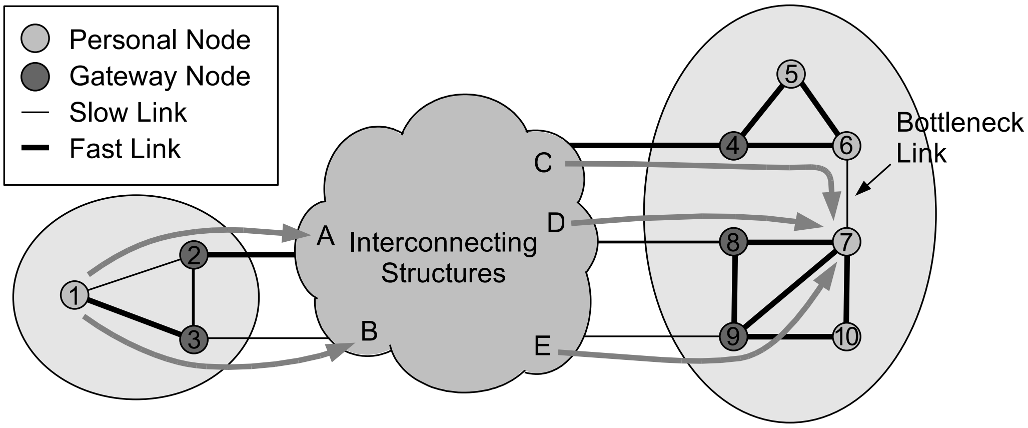


Figure 7.6: Cluster with bottleneck link

Gateway Node 2 inquires to the PN Agent about possibilities to connect to Node 7. Among the results, it selects which remote Gateway Node and TEP to use. This can be based solely on the QoS information about the TEP, which is the simplest solution. If so, it only picks the TEP with the best QoS parameters that can be used to connect to the destination Cluster. However, there are situations where this may not be sufficient. Examples include cases where there is a poor quality link in a Cluster. For instance in Figure 7.6, Nodes 4, 5, and 6 are best connected via Gateway Node 4, while Node 7, 8, and 9 are best connected using Gateway Node 8 or 9 due to the poor bottleneck link between Node 6 and 7. To achieve this, some information about the intra-Cluster routing is needed, such as the expected QoS between the Gateway Node and the final destination Node. Gateway Node 2 can then use this information in combination with the TEP QoS to decide which TEP to use. It can see that path D is better than path C and E in Figure 7.6. As before, this requires the exchange of extra information and may lead to frequent and unnecessary routing changes if the link qualities fluctuate. Hence, this should be used with care so that a reasonable level of routing stability can be maintained.

7.6.2 Tunnel quality assessment

To achieve good end-to-end communication quality within a PN, it is necessary to monitor the quality of available access connections as well as the quality of the entire tunnels. Usually the access connections are the bottlenecks, which means that those are the most important to monitor.

Fortunately, most access network technologies, including UMTS, IEEE 802.11 [82][27], and IEEE 802.16e [89], already do monitor the link quality. It is part of the handover mechanism and in those cases, it is advisable to use those measurements. However, it must be remembered that those measurements are designed to choose the best access point (or base station) and hence rarely provide any further details on the quality. For proper path selection, we also need to know things like the expected transmission count (ETX) [47] and the data rate.

Quite some work has been done in this area under the term vertical handover or media independent handover (MIH). There is currently also a standardization effort in this direction under the IEEE 802.21 working group [90]. The target of that work is to enable mobile terminals to make an informed selection of the best access network. The solutions are based on enabling relevant information from the lower layers and from the Interconnecting Structures in a standardized way. PN inter-Cluster communication can leverage on these technologies, but also needs to take into account additional aspects, since a Cluster may have several Gateway Nodes. The capabilities of the Gateway Nodes and their connectivity within the Cluster are examples of additional aspects that also should be considered.

For technologies that do not yet support any access link quality measurements, other methods are required. Though not the most effective way, network layer solutions can achieve some additional information about the link quality. By pinging the first hop access router or another Gateway Node, it is possible to detect the quality of the used access network and an entire tunnel respectively. Both delay and packet loss can be detected. The main drawback is the overhead created by such a technique and the difficulty in detecting the link throughput.

When significant changes are detected in the quality, relevant nodes must be informed, such as the PN Agent and Gateway Nodes to which there are active tunnels. However, sending around minor quality updates create a lot of unnecessary overhead without any real benefit. The latest quality information can always be included in the periodic updates that are transmitted anyway. Only when a significant quality drop is detected on a link that is currently used by traffic with strict quality requirements, an extra update message can be justified. Otherwise, it should be enough to wait until the next periodic update.

7.6.3 PN-wide broadcasting

For some PN applications, there is also a need for PN-wide broadcasting in addition to unicast routing and Cluster-wide broadcasting. Examples include revocation of security keys when a Personal Node is compromised, configuration updates affecting the whole PN, and service discovery.

The main problem with PN-wide broadcasting is to avoid sending the broadcast message over the Interconnecting Structures more times than necessary. In a Cluster with more than one Gateway Node, there must be an agreement about who will forward the message where. One simple solution is for a Personal Node to first do a Cluster-wide broadcasting and in parallel

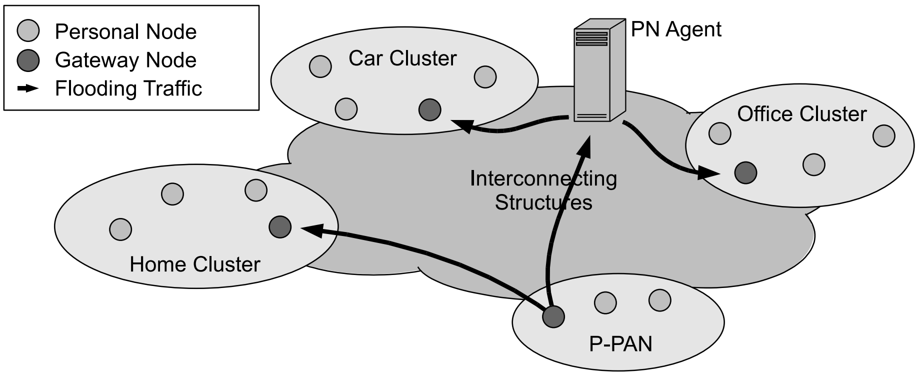


Figure 7.7: PN-wide broadcasting

send the message with unicast to the Gateway Node with the highest quality TEP for broadcasting to the other Clusters. Ultimately, this could be incorporated in the same packet. In this way, only that Gateway Node will take actions and thereby avoiding several Gateway Nodes doing the same job.

Furthermore, the best TEP in a Cluster may still not be fast enough to send the message to every remote Cluster. Sending the same message several times may cause too much delay and consume too many resources. Further, the Gateway Node may not have established tunnels to all the remote Clusters. Hence, we propose to send broadcast messages via the PN Agent since it has active sessions with all active Gateway Nodes. Either the PN Agent can do the entire job of sending the message to all other remote Clusters or the Gateway Node and PN Agent can do it together. The Gateway Node can deliver the packet over its active tunnels and then instruct the PN Agent to transfer the packet to the remaining Clusters on its behalf. Figure 7.7 shows this process assuming that the P-PAN Cluster only has an active tunnel with the Home Cluster. The quality of the involved TEPs should determine how much the Gateway Node should do itself and how much the PN Agent should do.

7.7 The Prototype

To demonstrate and better understand the mechanisms proposed in this chapter, the prototype presented in the previous chapters was extended. The main difference between this implementation and the main MAGNET prototype [125][76] is the use of Edge Routers. The MAGNET prototype relies on Edge Routers, while we do not use them at all. We implemented both the PN Agent and the Gateway Node functionality for inter-Cluster communication. In contrast to the intra-Cluster communication, we did implement encryption and data integrity protection for the inter-Cluster communication and signaling. While the inter-PN traffic is based on IPv6 as usual, only network access points and Interconnecting Structures using IPv4 are currently supported. All code was implemented in Python.

7.7.1 Python with libmcrypt

Python [189] is an object-oriented script language that has a very rich set of built in libraries. Python is available on all major platforms, including Linux. It is an easy to learn language that possesses most of the functionalities that can be expected from a modern programming language, such as a clean syntax, exception-based error handling, and high level dynamic data types. However, Python's perhaps best characteristic is that it allows one to quickly implement complex software and that was the reason for us to choose Python. Since Python is a kind of script language, it may not give us the fastest implementation, but should still be able to perform well.

For the encryption parts, we used libmcrypt [117], mainly because it offers an extension module for Python. Since libmcrypt itself is not implemented in Python, but in C, these parts of the code should be fast.

7.7.2 PNDB and packet formats

The central part of both the PN Agent and the Gateway Node implementations is the PN database that holds information about known and relevant Gateway Nodes and their active TEPs. Figure 7.8 shows the design (in the unified modelling language) of this database, which we call PNDB. What is shown are the common parts shared between the PN Agent and Gateway Node, including all the kept states. We can clearly see the various entities, their attributes, and how they relate to each other. The implementation of PNDB is exactly as shown in Figure 7.8, which means a lot of sequential searches through the database. Obviously, a scalable solution would need to use more efficient data structures. However, this implementation should still perform well for our tiny test PNs.

The PNDB consists of a list of all known Gateway Nodes and their pairwise keys. For each Gateway Node, the PNDB may hold its active TEPs and the Personal Nodes belonging to its Cluster. As explained earlier, this data is complete in the PN Agent's PNDB. For the PNDBs in the Gateway Nodes, this information may be partial, but if present for one particular Gateway Node entry, it is complete and updated for that entry.

For the inter-Cluster communication and signaling protocol, we defined one common packet format that can carry three different message types. The key message carries security key information, the TEP Node list message carries TEP and cluster member node information, while the last message type carries encrypted user data. Figure 7.9 shows all the packet and message formats in detail.

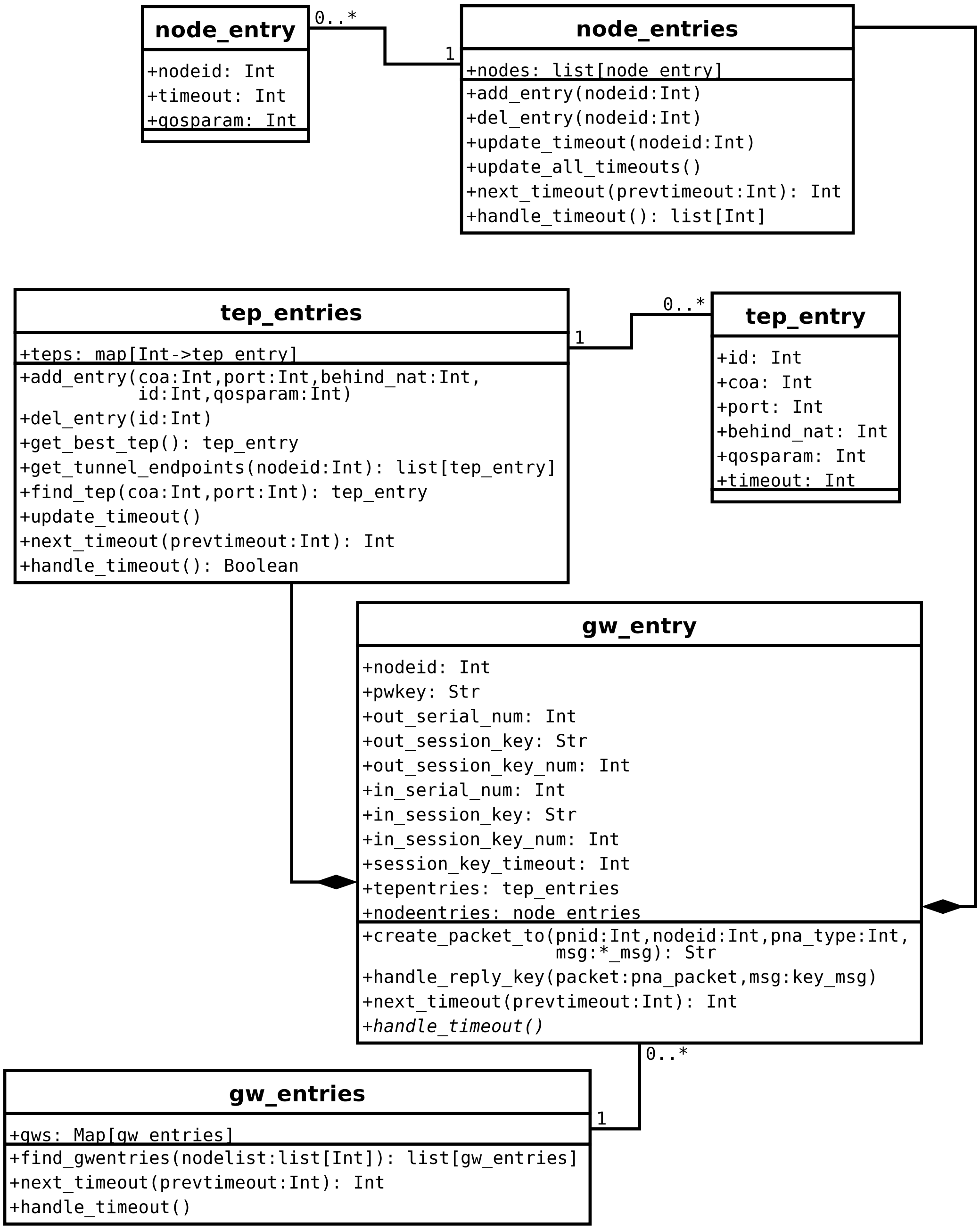

node_entry
+nodeid: Int
+timeout: Int
+qosparam: Int
0..*
1
node_entries
+nodes: list[node_entry]
+add_entry(nodeid:Int)
+del_entry(nodeid:Int)
+update_timeout(nodeid:Int)
+update_all_timeouts()
+next_timeout(prevtimeout:Int): Int
+handle_timeout(): list[Int]
tep_entries
+teps: map[Int->tep_entry]
+add_entry(coa:Int,port:Int,behind_nat:Int, id:Int,qosparam:Int)
+del_entry(id:Int)
+get_best_tep(): tep_entry
+get_tunnel_endpoints(nodeid:Int): list[tep_entry]
+find_tep(coa:Int,port:Int): tep_entry
+update_timeout()
+next_timeout(prevtimeout:Int): Int
+handle_timeout(): Boolean
1
0..*
tep_entry
+id: Int
+coa: Int
+port: Int
+behind_nat: Int
+qosparam: Int
+timeout: Int
gw_entry
+nodeid: Int
+pwkey: Str
+out_serial_num: Int
+out_session_key: Str
+out_session_key_num: Int
+in_serial_num: Int
+in_session_key: Str
+in_session_key_num: Int
+session_key_timeout: Int
+tepentries: tep_entries
+nodeentries: node_entries
+create_packet_to(pnid:Int,nodeid:Int,pna_type:Int, msg:*_msg): Str
+handle_reply_key(packet:pna_packet,msg:key_msg)
+next_timeout(prevtimeout:Int): Int
+handle_timeout()
0..*
1
gw_entries
+gws: Map[gw_entries]
+find_gwentries(nodelist:list[Int]): list[gw_entries]
+next_timeout(prevtimeout:Int): Int
+handle_timeout()

Figure 7.8: The PN database structure

The encryption is modeled after IPSec ESP but somewhat simplified (e.g., no SPI). The key messages are encrypted and signed with the pair-wise keys. They are used to agree and communicate the session keys between two Gateway Nodes or between a Gateway Node and the PN Agent. All other messages are encrypted using those session keys. While this security implementation captures the most essential aspects of security, it may not be the best option. Other, more advanced schemes, such as [138], [141], and [101], should

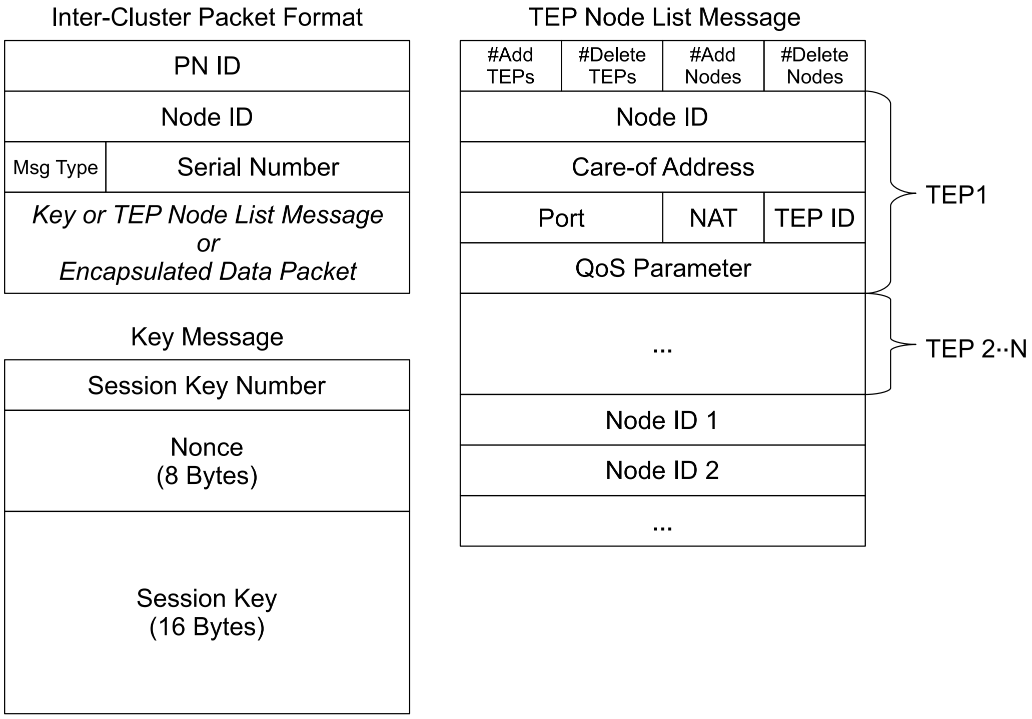


Figure 7.9: The inter-Cluster signaling packets

be considered for implementations in future PN products as they may offer better security and performance.

The TEP Node list message contains four lists that carry new TEPs, removed TEPs, new Nodes in the Cluster, and Nodes that left the Cluster. The message format is flexible enough to be used for all message types regarding TEPs and Cluster Nodes. It can be used for TEP updates, Cluster Node updates as well as queries and update subscriptions for such information. However, not all lists are used in all message types.

More message formats can be defined. Examples include new message types regarding NAT traversals and data packet forwarding through a third party. However, since support for NAT traversals has not yet been defined, we have not yet specified such message formats.

7.7.3 The PN Agent implementation

The PN Agent implementation is actually very simple. It only consists of the PNDB with minor extensions plus functionality for listening on a UDP port and handling received packets. The implementation is completely event driven and handles two types of events; timeout events in the PNDB and the reception of a UDP packet. Timeout events make sure that the PNDB stays up to date and that stalled information is cleaned up. A received UDP packet is first parsed, its signature verified, and then decrypted. The message inside

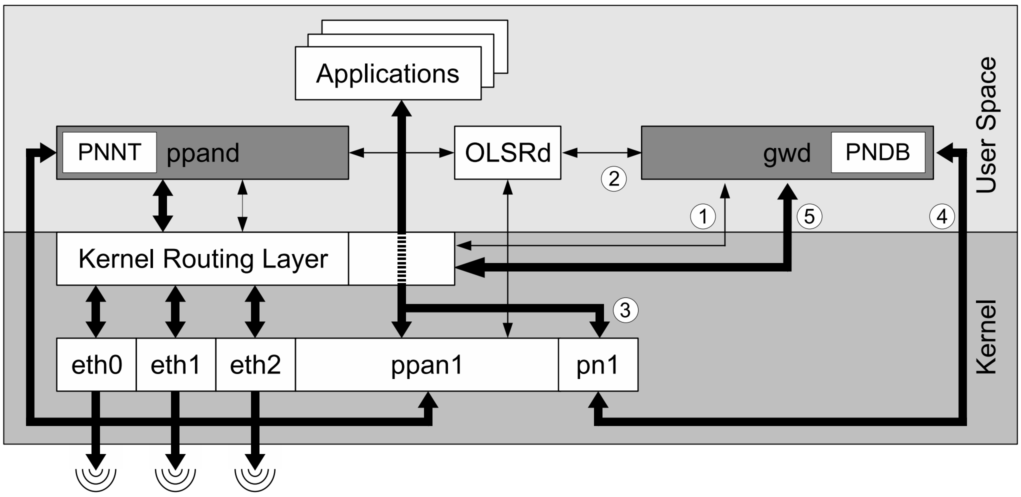


Figure 7.10: The Gateway Node implementation

the packet is then completely parsed and handled. The PNDB is updated with the new information carried in the message if any. Then, the PN Agent uses the information in its PNDB to formulate a reply or to inform other Gateway Nodes about the updates. Those messages are finally sent back encrypted using the session keys.

7.7.4 The Gateway Node implementation

The Gateway Node implementation is much more complicated. The main difference is that in addition to keeping the PNDB and handling signaling packets, it needs to do data packet handling and information gathering regarding available local TEPs and the Cluster Node membership. The latter two have only partly been implemented. Figure 7.10 shows how the implementation was done with the arrows indicated two not yet implemented and one only partially implemented. Despite these missing links, it has been possible to extensively experiment with the inter-Cluster communication implementation.

Except for some setup configuration scripts, all inter-Cluster communication functionalities were implemented in one program named gwd. Gwd is similar to ppand in the sense that it also creates a virtual interface (called `pn1` in Figure 7.10) and opens a UDP port for communication with other Gateway Nodes and the PN Agent. In addition to this, gwd also maintains the PNDB. It is responsible for all inter-Cluster signaling as well as data packet encryption and tunneling. Further, it needs to interact with the operating system to retrieve information about available network connections and their status. This information flow is, together with the inter-Cluster signaling traffic, indicated by arrow 1 in Figure 7.10. Arrow 2 indicates the information exchange between gwd and the OLSRd routing daemon. The

OLSRd routing daemon provides gwd with the Cluster member Node list, which it gathers from the intra-Cluster routing messages. In return, gwd provides OLSRd with quality parameters regarding its current connections to the Interconnecting Structures. OLSRd propagates that information through the whole Cluster so that every Node can use the best connected Gateway Node.

On a Gateway Node that has the best connection to the Interconnecting Structures, a data packet from an application on the Node itself or from other Nodes in the Cluster (also other Gateway Nodes) will be forwarded to the `pn1` virtual interface as indicated by 3 in Figure 7.10. The kernel will forward the packet to gwd (arrow 4) in the same way it forwards packets arriving on `ppan1` to ppand. If gwd has a valid tunnel that can be used for forwarding the packet to its final destination, it will encrypt and forward the data packet over the local interface associated with that tunnel according to arrow 5. The data packet is encapsulated in a UDP packet with the two TEPs of the tunnel as source and destination addresses and ports. If gwd does not have any tunnel that can be used, it will consult its PNDB for information on how it can establish such a tunnel. Since the PNDB may be incomplete, it may need to query the PN Agent. In that case, the data packet will be enqueued by gwd until further information is retrieved and a tunnel can be established.

The gwd application does not discover connections to the Interconnecting Structures on its own, but expects the operating system to do so. This decouples the PN functionalities from the access networks discovery technologies. It minimizes the impacts on existing legacy applications and also makes it possible for the PN communication to leverage the current and future developments in access networks.

Currently, most operating systems use DHCP [54], IPv6 router solicitations and advertisements [158], DNAv6 [157], or similar techniques for access network discovery. The most commonly used technique, DHCP, is typically handled by a user space program that reconfigures the routing table and other network-related settings in the operating system. Unfortunately, most DHCP clients will replace routes instead of keeping all alternatives. Because of this, we need to slightly modify the DHCP client.

Further, gwd needs to be informed when a new access network is discovered or an existing one disappears. On Linux, network settings like these can be accessed through the netlink socket interface [71]. Through that interface, user space applications, such as gwd, can retrieve and modify the network interface table as well as the current routing table. Furthermore, it is possible to subscribe to update events for those tables, which gwd should do. It is then updated as soon as a DHCP client finds a new connection and may then update its TEP list based on the new information. However, all of this together with connection quality detection has not yet been implemented in our prototype.

7.7.5 Lessons learned

The Python implementation of the inter-Cluster communication measured a total of 3200 lines of code (LoC), including comments. That includes 625 LoC for the PN Agent implementation, 1725 LoC for the Gateway Node functions, and 850 LoC for the common PNDB parts. Since this code was written in Python, which usually requires less LoC for the same amount of functionality, it is difficult to compare with the ppand implementation in Chapter 4, 5, and 6. However, the total code for the gwd implementation, which includes the most crucial parts of the inter-Cluster communication functions, is quite moderate.

This time around, we noticed performance degradation when using inter-Cluster communication. The round trip time increased by 1.5 ms when measuring between two Gateway Nodes using the same laptops (Intel Celeron M 1.6 GHz with 512 MB RAM) as in the earlier prototype experiments. Also the CPU load increased which caused the throughput to significantly degrade. A large part of the performance degradation can be attributed to Python. Since the per-data packet handling of gwd is not heavier than that of ppand, the two should perform about the same. Hence, we strongly believe that an optimized gwd implementation in C will achieve similar performance as ppand. However, with encryption, the throughput dropped even further while the delay increased by an additional 20 ms. Since the encryption implementation is done in C, this performance degradation cannot really be attributed to the non-optimized implementation. Instead, it demonstrates the importance of using hardware encryption.

We tried the same set of applications as we did for the intra-Cluster communication. Still, all worked without modifications, which should not be a surprise. While an extra round trip delay of 1.5 ms is hard to notice, we could easily notice the throughput degradation when performing large file transfers. Mobility events were hardly noticeable as long as only soft handovers between two different interfaces took place. Hard handovers, where an interface changes from one access point to another, will certainly be noticeable. Significant handover delay will be introduced, but most of it consists of link layer handover latency and DHCP reconfiguration latency. Due to the implementation limitations explained earlier, we were not able to test this.

One last requirement not yet addressed by this implementation is robustness to loss of signaling messages. In our test environment, virtually no packet loss occurred. In real Interconnecting Structures, packet loss may occur and this sometimes requires signaling messages to be retransmitted. The only real measure against this problem taken by our implementation was the priorities introduced on the interface queues. Signaling traffic was given higher priorities than other traffic, avoiding signaling packets to be dropped at the Gateway Node itself. This was done in exactly the same way as was done for ppand.

7.8 Summary

This chapter focused on communication between a person's Clusters. To implement this, secure intra-Cluster tunnels are established and maintained. The purposes of the tunnels are to protect the intra-PN traffic as well as transparently handle Cluster mobility. That is, the encrypted inter-Cluster tunnels are dynamic and updated when a Cluster changes its attachment point to the Interconnecting Structures.

Depending on the amount of infrastructure support that can be expected, we proposed several approaches to inter-Cluster tunneling. The best tradeoff seems to be to rely on a server located somewhere in the Interconnecting Structures. This server, we refer to as a PN Agent and is constantly updated with the current locations of the Clusters and how they can be contacted. When tunnels must be established or updated due to mobility, the Gateway Nodes in the Clusters may query the PN Agent for assistance. Enough information is then shared to effectively establish or maintain the tunnels.

Other approaches were also studied, such as a solution where access routers in the Interconnecting Structures can support the mobile Clusters with PN-specific functionality. We also sketched a solution where no extra support from the infrastructure can be expected, including entities such as PN Agents. For such scenarios, a PN is best implemented using peer-to-peer technology between the Gateway Nodes.

We also looked at the routing issue over the inter-Cluster tunnels. A routing approach integrated with the dynamic tunneling was proposed. It makes use of the PN Agent; the Cluster Node member lists are also communicated to the PN Agent. When a Gateway Node needs to transmit a packet to a Personal Node not in its Cluster, it may query the PN Agent to find out which tunnel to use or whether a new tunnel is needed. The benefits of this approach are mainly lower overhead and the ability to establish and maintain tunnels on demand.

An almost complete prototype of the proposed solution was made. The only missing functionality relates to the interaction with the operating system and the routing daemon with respect to retrieving the list of available access networks and the latest Cluster Node member list. The code was written in the Python programming language and showed the feasibility of the proposed inter-Cluster tunneling approach that we proposed.

Chapter 8

Foreign Communication

In the concept of PN, Nodes are divided into Personal Nodes and Foreign Nodes based on Trust Relationships determined by the owners. In the previous chapters, we have only discussed communication among the Personal Nodes themselves. Communication between a Personal Node and a Foreign Node, which we call foreign communication, is obviously also required. For many applications, a PN needs to interact with other PNs as well as PN-unaware devices. This includes using Services from other PNs as well as offering them Services. Whenever access to the Internet exists, a PN must be able to communicate with any Internet host, such as for surfing the web, reading emails, etc. It must also be possible to locate remote PNs when their locations are unknown and initiate communication with them. Hence, we need to enable foreign communication and that is the topic of this chapter.

Foreign communication is a topic very specific to PNs. The way a PN is defined makes it necessary to also define solutions for foreign communication. However, the foreign communication problem can be divided into two parts. One part concerns the mechanisms within the PN and those mechanisms are tightly coupled and specific to the PNs. The other part concerns the protocols between the edge of the PN and the Foreign Nodes. In the latter, existing standards are crucial. We cannot expect Foreign Nodes to understand one or a few specific protocols. Instead, we must build our solutions on widely adopted protocols. Given these limitations, the target of this chapter is to demonstrate that foreign communication is possible and how to achieve it.

This chapter is structured as follows. Section 8.1 introduces the most important requirements for foreign communication, while Section 8.2 lists related work. In Section 8.3, we start the description of foreign communication by investigating how to establish foreign communication. In Section 8.4, we go on to investigate how to connect the PN-internal networking with external networks, while we look at mobility aspects in Section 8.5. Authentication and anonymity for foreign communication is covered in Section 8.6 and in Section 8.7, we will discuss group foreign communication under an approach known as PN federations. Finally, in Section 8.8, we briefly look into how

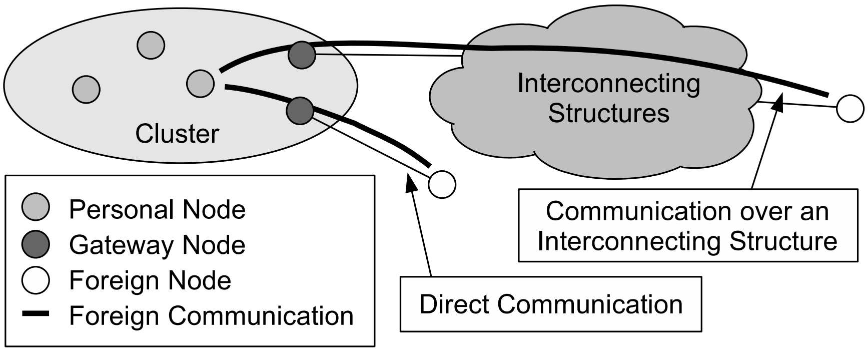


Figure 8.1: Types of foreign communication

our PN prototype could be extended with foreign communication capabilities, before we summarize in Section 8.9.

8.1 Requirements for Foreign Communication

Foreign communication requires special functionality at the Gateway Nodes [100][139]. It is namely the Gateway Nodes that will connect to the Personal Nodes with the Foreign Nodes. They need to bridge the mechanisms used within the PN with the mechanisms used outside. Figure 8.1 shows two examples of foreign communication; one with a Foreign Node using a common Radio Domain with one of the Gateway Nodes in the Cluster and one with a Foreign Node connected through an Interconnecting Structure. In both cases, a Gateway Node is involved in the establishment of the the end-to-end communication. Foreign communication may also need to use other types of networks to reach the destination device, such as multi-hop ad hoc networks. In either case, the Gateway Nodes need to understand and participate in the mechanisms of the external networks. Furthermore, they must be able to accept and handle connections initiated by Foreign Nodes as well.

For inter-Cluster communication, it is crucial that the following requirements are met:

1. Communication between applications on Personal Nodes and applications on Foreign Nodes must be possible. Both PN-aware and PN-unaware Foreign Nodes must be supported without requiring special functionality in the PN-unaware Foreign Nodes. Both Foreign Nodes and Personal Nodes must be able to initiate such communication.

2. Both Personal and Foreign Nodes can be mobile. Hence, mobility must be supported when either the Personal Nodes or the Foreign Nodes roam. Mobility of the Personal Nodes should also be handled when communicating with mobility-unaware Foreign Nodes. Furthermore,

the system should be able to find and select the best possible communication path at all times.

3. Just as the intra-PN communication, all foreign communication must be self-organized. Foreign communication must be established and maintained without support from the user. Furthermore, it should be able to automatically discover communication opportunities that the user may benefit from, both new connection possibilities and new interesting Foreign Nodes and their applications.

4. Foreign communication must work well even when the number of interactions between PNs and between PNs and PN-unaware nodes is large. Any solution must assist the user in managing these interactions.

5. Since foreign communication involves PNs and devices of other persons and organizations, security and privacy are even more important and difficult. Despite this, efficient and easy-to-use solutions are required that can handle the security risks caused by foreign communication.

8.2 Related Work

As mentioned earlier, foreign communication is a topic very specific to PNs. As a consequence, very little work has been published on this topic. However, the part concerning the protocols between the edge of the Personal Network and the Foreign Nodes needs well adopted standards to be truly useful. Therefore, we must investigate what is typical and what may become typical in the near future and build our solutions on those protocols. Since PNs are mobile, this mainly means that we need to look at what mobility solutions can be expected.

Chapter 7 already introduced most of the standardized mobility protocols, including Mobile IPv4 [178], Mobile IPv6 [103], and HIP [151][165]. However, with foreign communication, we can also consider mobility solutions at other layers than the IP layer, such as transport layer mobility protocols (e.g., [142][207][238][12]) and application layer mobility protocols (e.g., SIP mobility [203]). Unfortunately, none of the mobility support protocols can currently be described as widely adopted.

All these mobility protocols only focus on global and remote communication over Interconnecting Structures. There is however an alternative: contact networking [28]. Contact networking also considers local communication using link local addresses with direct neighbors. When local communication is not possible, it falls back on the Interconnecting Structures using techniques similar to Mobile IP. If direct local communication later becomes available again, it will switch back. Contact networking combines neighbor discovery, automatic addressing, link break detection, routing, vertical handover, mobility, and naming into one single solution. As such, it is highly applicable to PNs. However, contact networks requires all corresponding nodes also to be aware of contact networking, which means that a wide deployment is first needed. Unfortunately, there is currently no standardization or development effort going on in the direction of contact networking. Further limitations include the fact that there is no security and of course the inability to handle an entire PN instead of a single device. For these reasons we will not further consider contact networking for use with foreign communication. However, the use of the contact networking is a promising approach for foreign communication if it can become a widely deployed standard.

8.3 Setting up Communication with Foreign Nodes

For security reasons, it is important that foreign communication mechanisms remain separated from the intra-PN communication mechanisms. This means that the Gateway Nodes need to treat foreign communication in a different way than intra-PN communication and block all non-approved traffic from entering the PN. The Gateway Nodes must bridge the mechanisms used inside the PN and the ones used to communicate with the Foreign Nodes, since these are different and should never be combined.

As shown in Figure 8.1, the Gateway Node may connect with the Foreign Nodes in several different ways. Each way has its own mechanism and hence requires different approaches:

Direct communication At the connectivity level, the Gateway Node must of course share a common Radio Domain with the Foreign Node for direct communication. At the network layer, it needs to establish a network connection to the Foreign Node, which may be ad hoc and temporary. Temporary link local addresses can be used in the foreign communication as long as they are unique among the communicating peers. Another option is to use an already deployed network (such as a WLAN hotspot), if available, where addresses usually are assigned automatically (e.g., by DHCP [54]). At this point, communication can take place between any Node in the PN and the Foreign Node through the Gateway Node.

Communication over Interconnecting Structures If a Personal Node wishes to communicate with a Foreign Node that is connected through an Interconnecting Structure, the Gateway Node that links the Cluster up to the Interconnecting Structure needs to bridge the PN-internal mechanisms with the mechanisms used in that Interconnecting Structure. In this case, there is also the possibility of using the PN Agent as that bridge.

Communication over other network types Foreign communication may also need to use other types of networks, such as multi-hop ad hoc networks. The Gateway Node needs to understand and participate in the mechanisms of the external network. Consequently, Gateway Nodes may need to support several different network types.

It is also important to be able to switch between the different communication approaches when better alternatives arise or existing ones disappear [28]. More on this topic will be covered in Section 8.5 when we discuss mobility.

8.3.1 Foreign Node discovery

The first step in establishing foreign communication is to discover potential Foreign Nodes to communicate with. This step is called Node discovery and it is the task of the potential Gateway Nodes, as they are the only Nodes that can interact directly with the external networks. Gateway Nodes should keep a list of Foreign Nodes (or networks) so that foreign communication can be established when and if required. The Gateway Nodes inform the other Nodes in the Cluster, or the whole PN, about the Foreign Nodes (or foreign PNs) by broadcasting this within the PN as part of the routing protocol. At the service level, the Gateway Node may also discover services on local Foreign Nodes and advertise these within its Cluster. This can be done via populating the Service Management Node (SMN) in its Cluster [134][66] as discussed in Section 3.3.3. If the external network is PN-aware and has a SMN, it may interact with it.

To enable communication to remote Foreign Nodes, the Gateway Nodes advertise its current Interconnecting Structure connections. This advertisement can be done as a simple default gateway (or network prefix) within its Cluster. In this way, every Node in the Cluster knows which Gateway Nodes can be used to connect to the Interconnecting Structure.

Once a Foreign Node has been discovered and a Personal Node wishes to communicate with it, a Gateway Node must first be selected. If the Personal Node itself is a Gateway Node and has a link to the Foreign Node, it should choose itself instead of relying on other Nodes. If this is not possible, or not desirable (e.g., its own connection is limited or costly), it may choose to use another Gateway Node instead. It may be possible to choose from more than one Gateway Node. In some cases, such as in Figure 8.2, a direct connection (A) and several infrastructure-based connections (B, C, and, D) are possible at the same time. The Node needs to carefully select one of them, since the traffic between the two end Nodes has to go through the selected Gateway Node. In many cases, it is very hard to change Gateway Node without tearing down the connection and establishing a new one. States need to be transferred between the two Gateway Nodes and the Foreign Node may not support mobility. A last option is to use the inter-Cluster tunnels and choose

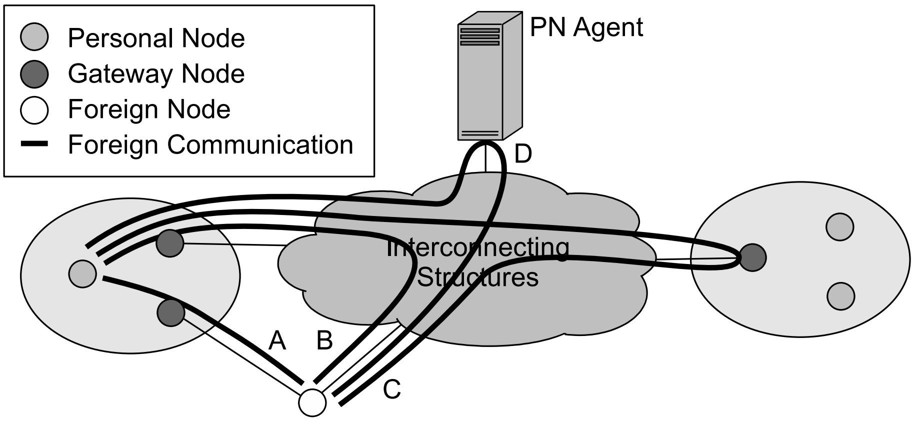


Figure 8.2: Four Gateway Node options

a stable Gateway Node (e.g., C) in a stable Cluster or the PN Agent (D). In that case, mobility is handled internally by the PN itself as described in Chapter 7, but at the same time, the routes are not optimal.

8.3.2 Accepting connections from Foreign Nodes

In some scenarios, it is interesting to consider the case where a Foreign Node wants to initiate a connection to the PN. If there is a direct connection, the Foreign Node can just initiate a connection to the present Gateway Node, which needs to handle it and establish an end-to-end session with one of the Personal Nodes within the PN.

When no direct connection exists, the Foreign Node needs to use the Interconnecting Structure. However, the Nodes of a PN can be mobile and change their point of attachment and therefore also their addresses used in the Interconnecting Structure. The only entity that does not change address is the PN Agent. The PN Agent is therefore an excellent point of contact for Foreign Nodes that wish to establish communication with a PN. It is only necessary to remember the address of the PN Agent to be able to initiate connections with that PN. To simplify the process even further, the address of the PN Agent can be given a name that can be resolved through DNS [148]. The PN Agent will know the location of all Clusters in its PN and can tunnel the packets to the final destination within the PN. At the same time, it will bridge the Interconnecting Structure and the intra-PN mechanisms.

8.4 Bridging Inside and Outside Protocols

After a Foreign Node has been discovered, it is time to establish data communication. The Gateway Node participates in the external network wherein the Foreign Node exists. If the Gateway Node itself wishes to communicate with the Foreign Node, nothing extra is required. The Gateway Node uses the mechanisms specified by the external network and acts as a normal node in the external network. However, if another Personal Node in the PN wants to communicate with the Foreign Node, then the Gateway Node needs to bridge the two network types.

There are two different ways a Gateway Node can connect the PN to the external network. End-to-end IP connectivity can be established between the two communicating endpoints using a network address translator (NAT) in the Gateway Node. An alternative is a Service Proxy that bridges the intra-PN and the external network. Section 8.4.1 discusses the network abstraction level solution using NATs, while Section 8.4.2 covers the Service Proxy solution at the service abstraction level.

8.4.1 At the network abstraction level

First, the source Personal Node needs to send packets to the destination Foreign Node such that they go through the selected Gateway Node. To do this, the source Node cannot just put the address of the destination node as the destination address of the packet. The intra-PN routing mechanism should not need to bother with every Foreign Node address as they may overload the intra-PN routing tables. Further, if two Gateway Nodes have a path to the same Foreign Node, it is not guaranteed that the packets will go through the selected Gateway Node all the time. The routing protocol may decide to switch Gateway Node at any time without prior warning. Finally, the address space may overlap and two different Foreign Nodes may use the same address. This can happen if the Foreign Nodes use RFC1918 addresses [192], addresses with link local scope [73], or if some nodes want to sabotage the PN operation by deliberately choosing non-unique addresses. This leaves three options for the Personal Node to make sure that the packets go through the selected Gateway Node:

Tunneling An IP-in-IP tunnel is established between the Personal Node and the selected Gateway Node. The endpoints of this tunnel are intra-PN addresses while the packets in the tunnel use the Foreign Node address as the destination. To send packets in the other direction, from the Gateway node to the Personal Node, no special functionality is necessary.

Source routing The intra-PN address of the Gateway Node is used as the destination address and the final destination, the address of the Foreign Node, is placed in an option field of the IP header (e.g., using a Type 0 routing header in IPv6 [48]). In principle, this scheme and the tunneling scheme are the same. However, source routing creates slightly less overhead.

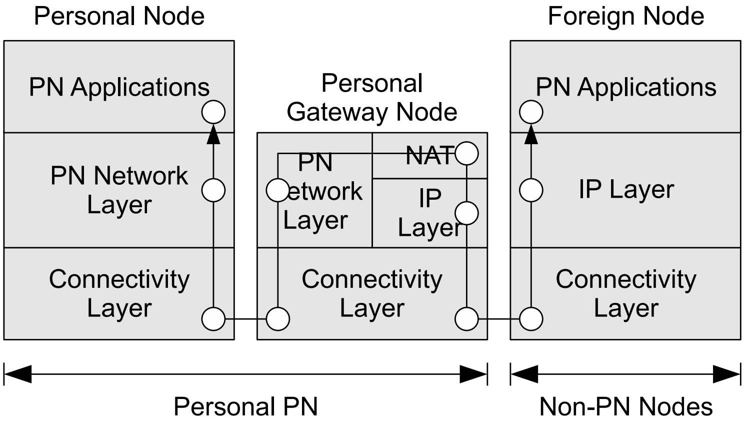


Figure 8.3: Foreign communication with a PN-unaware node

Address aliases We cannot use the addresses of the Foreign Nodes as mentioned above, but we can inject address aliases into the intra-PN routing protocol. If each Foreign Node and Gateway Node pair has a unique address alias within the PN, then a Personal Node can select both the Foreign Node and the Gateway Node to be used and hence, the control remains with the source Node. To minimize the burden on the intra-PN routing protocol, only aliases of (Foreign Node, Gateway Node) pairs that are in use should be injected.

While the Personal Node connects to the Gateway Node, the Gateway Node also needs to setup a path through the external network to the Foreign Node and install a state for the address translation so that intra-PN addresses can be translated to the addresses used outside. When all this is in place, traffic can start to flow. The Personal Node sends the packets to the selected Gateway Node, which will remove all PN-specific headers, encryptions, etc. Network addresses are then translated before being forwarded to the Foreign Node. To the applications, it will look like there is an end-to-end IP connection between the two end Nodes. Figure 8.3 shows an example of communication between a Node in a PN and a PN-unaware Foreign Node through a Gateway Node according to this scheme.

The address translation can be done with standard network address port translation (NAPT) technology, usually known as just network address translation (NAT) [209]. NATs are usually used to extend the limited address space in IPv4, but can also be used to hide internal address spaces used on private networks, such as PNs. The address space within a PN is to be used within the PN and is not globally unique. Because of this, network addresses must be translated and mapped when packets flow in and out of a PN and this is typical tasks of NATs. Hence, standard NAT technology will work.

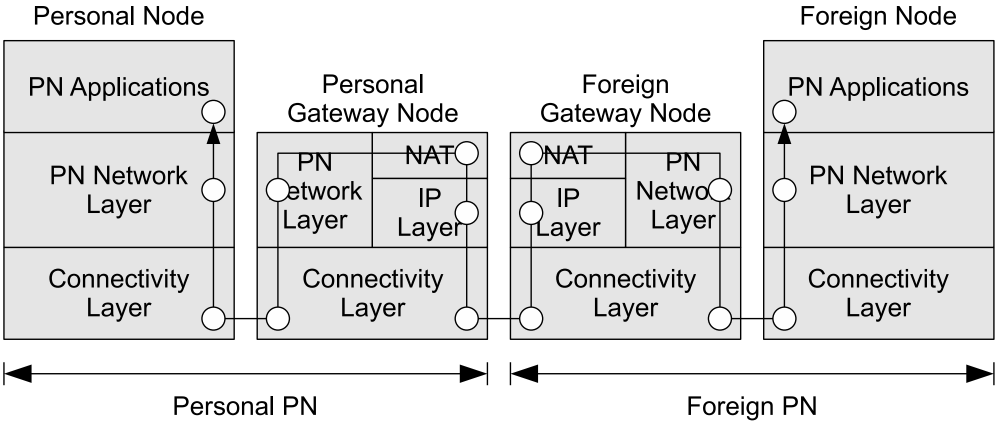


Figure 8.4: Foreign communication with a PN-unaware node

The next requirement regards accepting incoming connections from Foreign Nodes; it is actually not very different from the outgoing connections. First, the Gateway Nodes must somehow make sure that the PN is known to the outside world, for instance by advertising itself in the local vicinity. However, no PN-internal addresses should be advertised outside the PN because of privacy reasons and the fact that PN-internal addresses may only be unique within the PN. Instead, the Gateway Nodes advertise their own external addresses used within the external networks plus the Services that the Cluster or PN can offer. Then, as requests to establish communication arrive at the Gateway Nodes, they need to determine whether to accept the connection or not. If the connection is approved, an appropriate Personal Node is chosen to become the end Node. Higher layer information may determine this, based on what type of connection or what Service is requested.

From here on, there is no difference between this case and the case when a Personal Node initiated the connection. An address translation state needs to be installed in the Gateway Node that translates the network addresses. When packets arrive at the Gateway Node, it adds the necessary headers and encryption before the packets enter the PN and are forwarded to their final destinations. All the rest remains the same, except that the foreign PN will select the Gateway Node in this case.

The case when two PNs want to communicate with each other is nothing more than the combination of the two scenarios mentioned above: Figure 8.4 illustrates this.

8.4.2 At the service abstraction level

In many cases, it is actually not necessary to have end-to-end IP connectivity between the end Nodes. Another option is possible if we can assume that everything is provided as Services using a common service provisioning

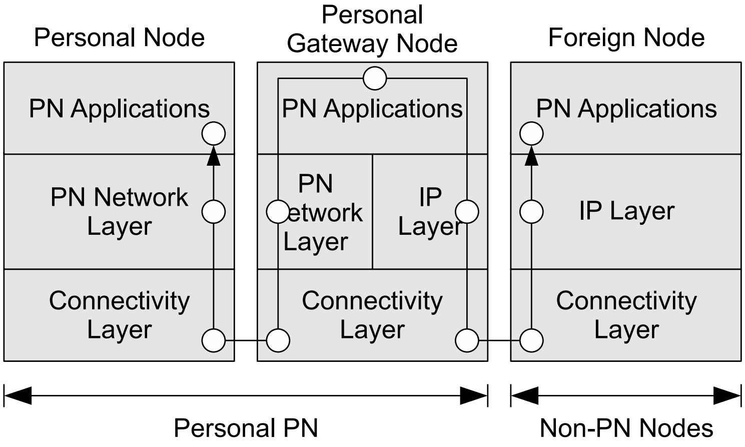


Figure 8.5: Foreign communication at the service level with a PN-unaware node

framework. Then the Gateway Nodes can use Service Proxies to relay Services outside the Cluster into the PN and vice versa. If a Gateway Node discovers a printing Service outside its Cluster, it can offer that Service to the Nodes inside its Cluster by starting a Service Proxy. A Client inside the Cluster can then use the outside printing Service by using the Service Proxy in the Gateway Node. The Service Proxy forwards the Service calls to the external printer Service and sends back the replies from the printer to the Client. In other words, it acts as a Server to the Client application and as a Client to the printing Service. The main purpose of the Service Proxy is to relay traffic in and out of the PN, which is necessary if this is not provided by the network level. Figure 8.5 shows this case. Note that there is no longer a need for NAT at the network level. Instead, there are two separate network connections; one from the application on the Personal Node to the Service Proxy on the Gateway Node and one from the Service Proxy to the application on the Foreign Node. The Service Proxy just connects the two.

The same solution can be used when a PN wants to offer Services to the outside world. The Gateway Nodes or the PN Agent can export these Services by means of similar Service Proxies. Such a Service Proxy exports the Services from the Personal Nodes and makes them available for Foreign Nodes instead. This solution may even be better than the network level solution from the point of view of security, since the Gateway Nodes and/or the PN Agent can control which Services are exported and which ones are not. A finer granularity of access control is also possible, since not all parts of the Service interface might be exported.

If we assume a service framework based on SMNs, as mentioned in Section 3.3.3 and the MAGNET architecture [134][66], then there is a SMN elected within the Cluster that manages the Services, Clients, and Service sessions. The SMN should also manage and coordinate the Service Proxies

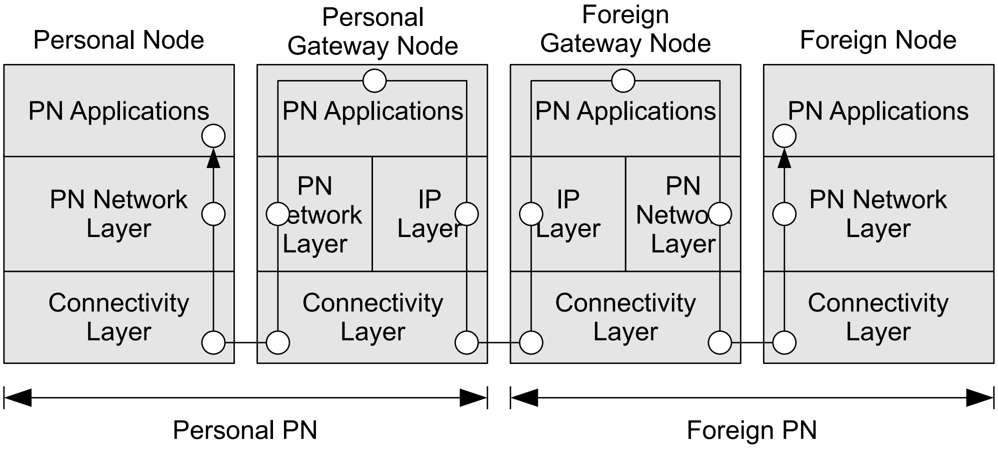


Figure 8.6: Foreign communication at the service level with another PN

on the Gateway Nodes and instruct them on what Services to export, import, and to what degree. The SMN may also control on which Gateway Node a particular Service Proxy for a particular Service should run. The fact that the Gateway Nodes and the SMN provide this functionality, also means that simpler devices can provide Public Services to the outside world without bothering about access control, authentication, etc.

Figure 8.6 shows the case where a Personal Node in a PN uses a Public Service offered by a Foreign Node in another PN. The Foreign Gateway Node exports a Public Service from one of the Nodes in its Cluster. Then the Personal Gateway Node detects this Service and offers it to the Personal Node using a Service Proxy. In this case, there is a difference between the Service Proxy that relays external Public Services into the PN and the Service Proxy that exports Public Services to the outside world. The former tries to secure the Service usage by using certificate servers or other similar methods, while the latter implements access control and authorization mechanisms. Both Service Proxies could operate under the control of their respective SMNs in the two PNs.

8.4.3 Network versus service abstraction level approach

A drawback of the Service Proxy approach is that we now have two connections instead of one at the network level. One connection is established between the Client and the Service Proxy and another one between the Service Proxy and the Service. If two PNs are communicating, there might be a third one between the two proxies running on each of the Gateway Nodes as shown in Figure 8.6 and this may adversely affect the QoS of the end-to-end communication.

It is of course possible to use both the network level and service level solutions at the same time. QoS-sensitive communication might use the network level mechanisms to establish end-to-end IP connectivity between the end Nodes, while others may use the Service Proxy approach. One option is to use the network level mechanisms for Personal Nodes using Services outside the PN, while a Service Proxy is used when the same Personal Node exports Public Services to the outside. In this way, we will avoid having two Service Proxies and increase the possibilities for access control of the exported Services.

8.5 Mobility and Gateway Node Handover

Since many PN Nodes are mobile, it is natural that also foreign communication paths need to change. There are several reasons why mobility is required:

1. The Gateway Node switches its point of attachment to the Interconnecting Structure requiring a different care-of address (CoA).

2. Direct communication between a Personal Node and a Foreign Node is no longer possible due to mobility. Consequently, a switch to Interconnecting Structure-based communication may be required.

3. When direct communication becomes possible, it is usually better to switch from a connection via an Interconnecting Structure to a direct connection. Usually, a direct connection can offer better bandwidth, better QoS, and lower cost.

4. The selected Gateway Node becomes unavailable or loses its connection to the Foreign Node (or the Interconnecting Structure) and another Gateway Node must be used.

5. The Foreign Node might be mobile. However, in that case, it can be assumed that it has its own support for mobility.

These examples demonstrate the importance of good mobility solutions, also for foreign communication, as we would like on-going sessions to proceed without interruption. To overcome this problem, we propose two solutions. The first one is very simple, but non-optimal. It relies on always using Interconnecting Structures and routing the traffic through the PN Agent, which is a fixed node with a fixed address. The second solution is more complex but achieves much better routing, since traffic is routed through the most appropriate Gateway Node. In this case, mobility of the Gateway Node (terminal mobility) as well as a switch to another Gateway Node must be supported.

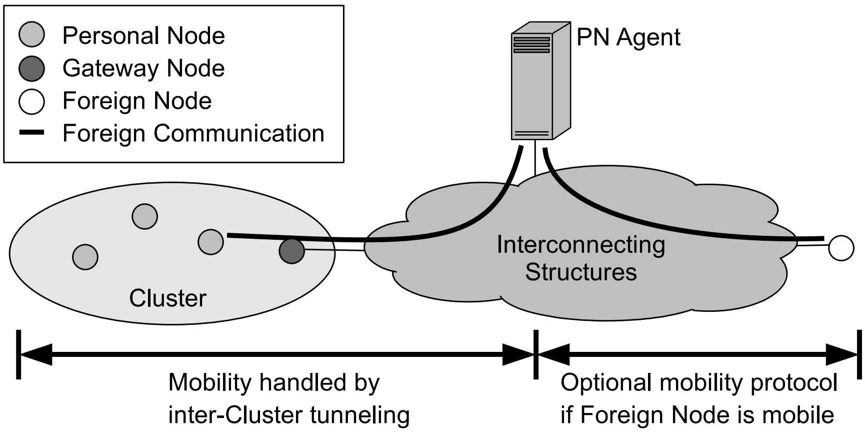

Personal Node
Gateway Node
Foreign Node
Foreign Communication
PN Agent
Interconnecting Structures
Cluster
Mobility handled by inter-Cluster tunneling
Optional mobility protocol if Foreign Node is mobile

Figure 8.7: Foreign communication at the service level with another PN

8.5.1 Always using the PN Agent

The main advantage of this solution is its simplicity. All foreign communication goes through the PN Agent, which means that only the PN Agent needs to implement the bridging as shown in Figure 8.7. The PN Agent never changes its address so there is no need for any external mobility solution between the PN Agent and the Foreign Nodes unless the Foreign Node is mobile. In addition, the intra-PN routing will be very simple. Packets with a Foreign Node destination are forwarded to a Gateway Node and then over the Interconnecting Structure to the PN Agent. A default route within the PN can achieve this. Hence, there is no need for source routing or tunneling to make sure the packets arrive at a particular Gateway Node.

Obviously, this solution has many limitations. To route all foreign traffic (both directions) through the PN Agent leads to non-optimal routing and a potential bottleneck. Furthermore, direct communication to a Foreign Node is not at all possible. On the other hand, this solution handles mobility of the PN Clusters also when communicating with Foreign Nodes. The link between the PN Agent and Foreign Node remains stable and does not change unless the Foreign Node is mobile. All mobility of the Personal Nodes takes place within the PN and is handled by the dynamic tunneling provided by the PN, which can be extended to include the PN Agent. The Cluster Gateway Nodes inform the PN Agent about their care-of addresses (CoAs), including changes thereof. This enables the Personal Nodes to be mobile while communicating with Foreign Nodes and this only at the cost of non-optimal routing on the Interconnecting Structure side.

This solution is similar to MobileIPv4, when using reverse tunneling as described in [150], except that foreign agents are never used in our case. While it would be possible to actually use Mobile IP, it would mean using two mobility solutions in parallel, which is of course unnecessary. Hence, it is better to rely on the mobility mechanisms already provided by the PN.

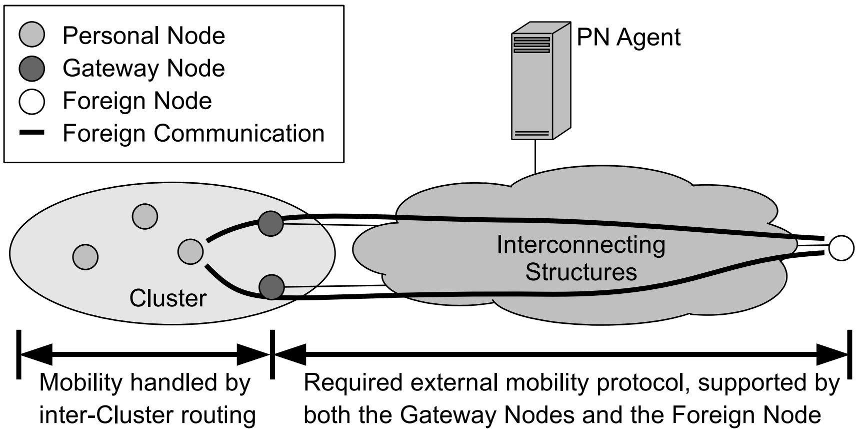

Personal Node
Gateway Node
Foreign Node
Foreign Communication
PN Agent
Interconnecting Structures
Cluster
Mobility handled by inter-Cluster routing
Required external mobility protocol, supported by both the Gateway Nodes and the Foreign Node

Figure 8.8: Foreign communication at the service level with another PN

There is one more important reason why this solution is good and that is privacy. If the traffic goes directly from the Gateway Node to the Foreign Node, the CoA of the Gateway Node will be known to the Foreign Node and this address can reveal the user's current location. If the traffic goes via the PN Agent, this address will be hidden from the Foreign Node, thereby guarding the location privacy of the user.

8.5.2 Using the optimal Gateway Node

To enable mobility with optimal routing for foreign communication, two problems must be handled. First, address changes and multi-homing at the Gateway Node must be handled. Second, support for switching between two Gateway Nodes (or the PN Agent) must also be available. To sustain the ongoing connections, both require some sort of mobility support between the Gateway Node and the Foreign Node as shown in Figure 8.8. Hence, a common external mobility protocol is needed. In addition to this, a Gateway Node handover also requires the two Gateway Nodes to exchange state information.

Consequently, there is a need for an intra-PN protocol that can communicate the intention to change Gateway Node and then transfer these states between the two Gateway Nodes. The protocol should preferably be able to act before the old Gateway Node looses its connectivity or becomes unavailable. The protocol must also trigger the external mobility protocol to take appropriate actions.

It cannot be assumed that Foreign Nodes can handle a sudden change of address at the Gateway Node without mobility support. Most current IP nodes on the Internet or elsewhere cannot handle such changes without loosing the connection. A widely deployed mobility standard is required; otherwise, the chance that a Foreign Node actually implements a proper solution is very slim. With this in mind, there are only a few options that can handle mobility between the Gateway Node and the Foreign Node:

Mobile IPv4 [178] This is a well-established protocol for mobility on IPv4 networks. The PN Agent can act as home agent (HA) and the Foreign Node can use IPv4 as usual without any modifications. Whenever a Gateway Node changes address or another Gateway Node is selected, the HA at the PN Agent is informed. However, the Foreign Node cannot be informed, which is a limitation of Mobile IPv4 as it is now being standardized. Consequently, the Foreign Node will always send its packets via the HA (PN Agent), which is still non-optimal. Only packets from the Personal Node to the Foreign Node will take the direct path. Furthermore, Mobile IPv4 is not able to switch to direct local communication when such possibilities exist. Instead, all traffic has to go via Interconnecting Structures. In the end, the benefits of this option, compared to always routing through the PN Agent, is rather limited.

Mobile IPv6 [103] The most important difference between Mobile IPv4 and Mobile IPv6 is the use of binding updates (BU), which are the messages sent to the HA to update the CoA. In Mobile IPv6, BUs are also sent to the Foreign Node. If the Foreign Node implements IPv6 and Mobile IPv6, it can directly send its packet to the Gateway Node instead of via the HA (i.e., PN Agent). Otherwise, the two protocols work in the same way and have the same limitations.

Host Identity Protocol (HIP) [151] HIP is also able to handle mobility [165], though it currently seems that it too lacks support for local direct communication. However, HIP is not yet a standard and still has a long way to go before becoming one. It is therefore unlikely that we can expect Foreign Nodes to implement HIP in the near future. However, if the Foreign Node supports HIP, then this could be a good choice.

Transport and application layer mobility protocols As explained earlier, many proposals have been made for mobility handling at the transport layer [142] [207] [12] [238] and at the application layer [203]. However, all of them have the same problem as HIP; none is a standard yet and none has any real deployment. However, if any of them should take off and become widely deployed; they may all be good candidates.

Contact networking [28] The most important benefit of contact networking compared to the previous alternatives is its support for local direct communication. This perhaps makes contact networking the best option as long as it is widely deployed. However, no standardization effort is currently under way regarding this proposal. Furthermore, the security aspects are still to be addressed.

If two PNs are communicating, there is one more possibility. It would be possible to deploy special functionality at the Gateway Nodes that handles mobility between the PNs. In the case that any of the Gateway Nodes needs to change network address, a special inter-PN mechanism could be used to maintain the connection. They could exchange the addresses of their PN Agents to fall back on, in case the current communication link breaks. In addition, the address changes should be communicated directly between the foreign Gateway Nodes in the same way as in the protocols above.

In an ideal world, we conclude that a solution based on a combination of HIP and contact networking is probably the best solution. Such a solution should be able to handle both security and mobility across both direct connections as well as Interconnecting Structures for all types of foreign communication.

Another important aspect of Gateway Node mobility is the handover delay. Coordination among the Personal Nodes and the Gateway Nodes, selection of a new Gateway Node, transfer of states, and the operations of the external mobility protocol all introduce delay. If a connection carries real-time data, serious quality problems may arise if the time to adapt becomes too long. Acting in advance, with a “make before break”-approach in combination with very speedy operations of all involved mobility interactions, is preferable.

8.5.3 Using Service Proxies

When using Service Proxies instead of NAT, it is still possible to use the same solutions for mobility between the Gateway Node and the Foreign Node. There is also the possibility to use service level mobility such as SIP mobility [203] instead of Mobile IP. Inside the PN, nothing at the network level needs to change.

The only additional problem that needs to be handled is the change of Service Proxy. Imagine that a Gateway Node running a Service Proxy is about to become unavailable, then a change to another Gateway Node with the same type of proxy is needed. This would require similar handover procedures as in the network level solution. However, the amount of state information that must be transferred may be larger and more complex. Typically, a Service Proxy will keep more state information related to the Service itself. Perhaps buffered data or remote procedure calls as well. Except for this, a transfer from one Gateway Node to another will work in the same way.

8.6 Authentication and Anonymity

Before communicating with a Foreign Node, one should first verify the authenticity of that Node. This can take place as soon as the Node is discovered and before being announced within the PN in order to avoid overloading the

PN with information about non-trusted Nodes. However, the authentication operation may be quite heavy and consequently it may be better to perform it on demand instead. In either case, several authentication approaches exist, such as public key infrastructure (PKI) and reputation-based approaches.

Assuming that two PNs want to communicate, both PNs could contact a certificate server (CS) to get the necessary authentication information. A CS can be a different server supported by a third party or constitute a separate functionality in the PN Agents themselves. However, contacting a CS every time a Foreign Node needs to be authenticated is tedious and will not work when the CS is unavailable. Various caching mechanisms of the certificates can be used. The PN Agent can, for instance, keep certificates of known PNs on behalf of its PN.

Another way of authenticating foreign PNs is to have it performed manually by the user when two PNs meet physically. Imagine that two persons and their PNs meet in the street. By means of proximity authenticated channels or other types of secure channels, they can physically authenticate each other. A certificate can be exchanged as part of the authentication and this can be cached just as a certificate received from a CS. Further, these certificates can be exchanged between friends in a similar way to the web of trust as used in PGP [245].

When a foreign PN can be authenticated, it is also possible to establish a secure connection. This will make it impossible for other Foreign Nodes to interfere with the foreign communication between them. It also includes the protection of the data itself from wiretapping and manipulation.

These are just examples of how Foreign Node authentication can be done. Many more options exist in the literature. For more examples and further details on these topics, see [140].

Regarding anonymity, things become much more difficult when considering communication between multiple foreign PNs. The solution of having one common key for encryption of Node and PN identifiers that was proposed in Chapter 4 for protecting the intra-Cluster communication will not work. One cannot share a key with every friend one has without them also sharing the same key with each other. Furthermore, the solutions proposed in Chapter 7 for securing the infrastructure-based connections will not work either. In those solutions, only one of the peers needs to remain anonymous. Hence, the infrastructure-based equipment announces its identifier and public key in the open so that clients anonymously can establish a secure link.

To allow a node to reveal its identity and establish a secure connection to another node only if there is a Trust Relationship and otherwise remain anonymous, seems impossible if the other node also has the same requirement. Not knowing the identity of a newly discovered device, and trying to figure out if there is a Trust Relationship with that device, are contradictory requirements. More efficient solutions than trying each available Trust Relationship, one by one, are difficult to find. Possible candidates may involve the two peers gradually give away clues of their identities, until either of them concludes that only one option remains, whereby that option is tried. If it succeeds, there is a Trust Relationship, otherwise not. For the time being, this problem remains an open and challenging research issue.

8.7 PN Federations

Not only bilateral communication between PNs is of interest, also group communication among several persons and their PNs is of importance. In PN federations (PN-Fs) [164][75], the PNs will start to interact and collaborate for the purpose of achieving a common goal. The federation may host cooperative services by using shared resources from the participating PNs. Some initial work has been done to define how such PN-Fs can be formed in reality [139][81]. Whatever way is chosen, one thing is for certain: foreign communication will be a crucial building block.

One approach to PN federation is to build a communication structure among the participating PNs using bilateral connections between the PNs. That is, the PNs in the PN-F establish foreign communication among themselves in one of the ways specified in this chapter. Over these connections, services are shared and traffic is routed. Then most of the problems regarding the PN-F usage phase is solved. However, to achieve a complete solution for PN-Fs, we also need methods for creation and destruction of PN-Fs, invitation and eviction of participating PNs, formation and adaptation of the PN-Fs, security and privacy, and much more. Several initial proposals have been made and are described in [139].

8.8 Prototyping Foreign Communication

Due to lack of time, our prototype is not yet capable of handling foreign communication. However, it would not be excessively hard to enable foreign communication always through the PN Agent as described in Section 8.5.1. It mainly requires updates to the intra-Cluster and PN-wide routing protocol implementation so that foreign communication traffic flows through the PN Agent. At the PN Agent, a simple type of NAT is required that translates between the intra-PN addresses and the addresses used in the Interconnecting Structures. To also accept incoming connection attempts requires further functionality, such as locating the final destination Node within the PN. Finally, to implement foreign communication through the most optimal Gateway Node with mobility support requires the most amount of work. This is due to the required coordination protocols, the external mobility protocol, and their interactions.

The "easy way" of implementing foreign communication is not what is most useful from a research point of view. With an extensive prototype, we will be able to discover what are the best alternatives for foreign communication based on clear questions, such as effectiveness, delay, overhead, etc. Eventually, we will be able to fully answer the questions about which solutions are the better ones. However, all this remains future work for now.

8.9 Summary

In this chapter, we looked at how communication with PN-unaware devices and Foreign Nodes can be done. Due to security reasons, the Gateway Nodes need to treat foreign communication in a different way and block all non-approved traffic from entering the PN. Intra-PN communication mechanisms must remain separated from foreign communication mechanisms. Hence, the Gateway Nodes must bridge the mechanisms used inside the PN with the ones used to communicate with the Foreign Nodes.

Two different solutions were described; either end-to-end network layer connectivity is established across PN boundaries using network layer address translation or a service proxy bridges between the PN and the Foreign Nodes outside.

Since many PN nodes are mobile, also foreign communication paths need to support mobility. We introduced two approaches to handle this: either always sending foreign traffic via the PN Agent and handling all mobility using the inter-Cluster tunneling mechanisms, or using the most optimal Gateway Node and then using a well-adopted mobility protocol between the Gateway Nodes and the Foreign Nodes. The first option is the simplest and works with all current Foreign Nodes as well as PN-unaware devices. The second option is more complex and requires the Foreign Nodes to implement a mobility protocol, but also allows for more efficient routing.

When group communication among several persons and their PNs is needed, PN federations (PN-Fs) may be used. One approach to PN-F is to let the PNs in the PN-F establish foreign communication among themselves using the foreign communication mechanisms specified in this chapter. Over these connections, PN-F services are shared and traffic can be routed.

The main unanswered questions with respect to foreign communication relate to the implementation and security. We would like to implement some basic foreign communication functionality so that the concepts and solutions can be properly tested and verified to work efficiently in a prototype. Regarding security, more work is required to make sure authentication and access control among PNs are both secure and easy to use. However, we believe that these open issues use solvable in the near future.

Chapter 9

Conclusions

In this thesis, we have first defined the concept of personal networks; a future concept of advanced personal communication. Today, wireless communication technologies are specialized towards particular communication problems in order to better address particular niche problems. This causes difficulties for the end-users that have to understand and master all these technologies and accept that they do not fully work together. Hence, the main problem for personal networks is how to make these technologies best complement each other and work together seamlessly.

We started out by showing that there is a real benefit from having communication among all one's electronic devices and that this can lead to new applications and services. Further, it will be easier for service providers to offer added-value services to a person if there is a personal communication platform to build on. Hence, the concept of personal networks is an attractive proposition with many applications.

In Chapter 2, we first gave the user requirements we believe a personal network must meet. Then, we investigated existing technologies that can be used to make the concept of personal networks a reality. Evidently, there are plenty of partial solutions that can solve individual problems, but no overall solution. Instead, the individual partial solutions are frequently incompatible with each other. There is a clear lack of concertation among technologies to reach the vision of personal networks.

Therefore, we set out to define such solutions. We started in Chapter 3 with defining an architecture for personal networks. We defined concepts mainly related to the networking aspects of personal networks. Among other things, we gave a precise and technical definition of what exactly a Personal Network is. Furthermore, we assessed the architecture's ability to meet the requirements we established as far as is possible for something that is not implemented, nor even fully specified. Several issues were discussed.

After that, each of the networking aspects of the architecture was worked out in the remaining chapters. In Chapter 4, we looked at local personal communication with the formation of Cluster of Personal Nodes. In Chapter 5, we studied broadcasting within a Cluster using flooding techniques. Unicast routing and link quality assessment were covered in Chapter 6. In Chapter 7, we addressed the issue of connecting a person's local Clusters using infrastructure networks in a seamless fashion. Communication between Personal Networks belonging to different persons are introduced in Chapter 8.

Together, Chapter 4 to 8 show the feasibility of the architecture. Together the solutions make up the networking parts of a personal network. In all, but the last chapter on foreign communication, a prototype has been developed that clearly demonstrates the possibility to build a real Personal Network. Also the foreign communication parts are not far from becoming a reality. In conclusion, this is the main contribution of this thesis. However, one question remains to be answered; are we there now?

9.1 Are We There Now?

Clearly, we have a prototype of a Personal Network. Furthermore, prototypes have been developed in the IST MAGNET and PNP2008 projects. Prototypes that also demonstrate other aspects than only the networking aspects. However, these are just prototypes, not ready products that end-users can acquire and start to use. Hence, personal networks is only a reality in the research labs. But how far are we with the personal networks technology? Is it ready for large scale deployment?

To answer that question, it is necessary to go back to the requirements and not only the high level architectural requirements in Chapter 2, but also all the requirements on the individual components of the architecture given in Chapter 4 to 8. Regarding the requirements on the architecture, several issues still remain as discussed in Chapter 3. Such issues include whether Personal Networks will be able to meet the security demands, whether the Personal Networks actually will be as easy to use as we hope, and whether Personal Networks can support the social interactions between users. Unfortunately, we believe there is only one way to find out and that is to build a prototype and test it with real users.

In the remaining chapter we have identified the following gaps:

1. We stressed the need for support of heterogeneous link layer technologies. This was done for most of the chapters, but rarely studied. Hence, more research is required regarding Personal Networks with multiple simultaneous link layer technologies. Especially in the area of unicast routing and flooding.

2. The multi-hop routing in Clusters is far from perfect. In the best case, the routing quality is satisfactory for the majority of applications. However, other requirements, such as low overhead and low power consumption have more or less been neglected up until now. Chapter 6, where

this was studied, is more of a research in progress than a finished solution. Hence, the work needs to continue with even more measurements before a final solution is complete.

3. Routing over infrastructure networks is an issue. The approach taken in Chapter 7 is to design a system that works over as many connection types as possible. However, many systems are not built for personal networks communication and will cause problems, such as during handovers. Furthermore, there is a lack of cooperation between the various alternatives, which may hamper handovers between technologies. Self-organized infrastructure connections, which automate the network discovery, authentication and association processes, are required.

4. Network layer anonymity and privacy is not what it should be. In a world where a person's devices communicate with each other, privacy becomes a greater issue. Anonymity is one of the few solutions to this problem. However, we have only been able to identify anonymity solutions for intra-PN communication, but not for the general inter-PN communication case. Furthermore, most link layer technologies need to be enhanced with better anonymity features.

There are also areas out of the scope of this thesis, where further research is required. Such areas include security, such as protection against denial of service attacks, viruses, and other threats. How to handle security and trust in an easy-to-use way regarding foreign communication. Another very important area regards application and service support and includes service discovery and management as well as context information management and awareness. Finally we need applications that are able to do what people want in an easy-to-use way.

9.2 Future Directions

The concept of personal networks is an ambitious endeavor. It certainly needs to go much beyond a single PhD thesis. The purpose of this thesis was only to propose and research technical solutions for the networking aspects and this has to a large degree been met. We believe that the remaining technical obstacles, such as the ones listed in Section 9.1, also can be worked out. However, this will not be sufficient for successful deployment of personal networks.

The real success of personal networks will only be achieved when most device manufacturers, network providers, and content providers offer personal networks enabled products and services. However, that requires far-reaching interoperability. To reach this, these players, which come from different parts of the industry, must start a collaborative effort and that requires good incentives for each of them. The personal network must not only be beneficial for the user as has mainly been the focus of this thesis, but also for manufacturers and providers.

For the network providers, benefits will come from increased communication needs in terms of more required bandwidth, the need to constantly be connected, and to be connected through more types of connections. Network providers that can offer not only one type of access, but a multitude of infrastructure-based connection types, such as UMTS, WiMAX, WLAN, Cable, and DSL, will have an advantage if they can provide one single seamless and well-concerted network service.

Content providers will mainly benefit from a unified platform through which they can offer their services to many more users in a homogeneous and secure way. The mobility aspects of personal networks will make it possible to almost always reach their customers with their services.

Device manufacturers will probably face they biggest challenges, since it involves industries so diverse as manufacturers of consumer electronics, mobile products, PC equipments, home appliances, home automation, and many more. In the short term, manufacturers that can offer devices capable of participating in people's PNs will have a leading edge. In the longer term, we believe that personal networks will spur new sorts of applications and that requires new types of electronic devices.

The work must now also start as an industrial effort. Standardization, cross-industry liaisons, and alliances that can safe-guard a smooth and interoperable development of personal networks are necessary. It is certainly a positive sign for personal networks that standardization and other businness-related efforts are starting to be discussed within both IST MAGNET Beyond [130] and Freeband PNP2008 [180].

Appendix A

List of Abbreviations

3GPP	Third Generation Partnership Project
AHBP	Ad Hoc Broadcast Protocol
AIPN	All-IP networks
AN	Ambient Networks
AODV	Ad Hoc On-Demand Distance Vector
ARQ	Automatic Repeat Request
BAN	Body Area Network
BRG	Broadcast Relay Gateway
BU	Binding Updates
CA	Certificate Authority
CAN	Community Area Network
CBB	Counter-Based Broadcasting
CB-PFS	Counter-Based PFS
CCA	Clear Channel Assessment
CHARM	Channel-Aware Rate Adaptation Algorithm
CoA	Care-of Address
CS	Certificate Server
CTS	Clear to Send
DAD	Duplicate Address Detection
DCF	Distributed Coordination Function
DHCP	Dynamic Host Configuration Protocol
DME	Device Management Entity
DNA	Detecting Network Access
DNS	Domain Name System
DoS	Denial-of-Service
DSDV	Destination-Sequenced Distance-Vector Routing
DSL	Digital Subscriber Line

DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
DYMO	Dynamic MANET On-demand Routing Protocol
E-CDS	Essential Connecting Dominating Set
ECN	Explicit Congestion Notification
ELN	Explicit Loss Notification
EMNUN	Estimated Maximum Number of Uncovered Neighbors
ER	Edge Router
ETT	Estimated Transmission Time
ETX	Estimated Transmission Count
EWMA	Exponentially Weighted Moving Average
FA	Foreign Agent
FEC	Forward Error Correction
Fednets	Federation of Networks
FIFO	First In First Out
FMIPv6	Fast Handover for Mobile IP
FP6	Sixth Framework Programme
FSP	Flooding with Self-Pruning
GLL	Generic Link Layer
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HA	Home Agent
HI	Host Identifier
HIP	Host Identity Protocol
HMIPv6	Hierarchical Mobile IPv6 Mobility Management
HSDPA	High-Speed Downlink Packet Access
i3	Internet Indirection Infrastructure
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineers
IP	Internet Protocol
IPC	Inter-Process Communication
IrDA	Infrared Data Association
ISTAG	Information Society Technologies Advisory Group
IST	Information Society Technology
LLAL	Link Layer Adaptation Layer
LoC	Lines of Code
LQA	Link Quality Assessment
MAC	Medium Access Control
MAGNET	My Adaptive Global Net
MANET	Mobile Ad Hoc Network
MIH	Media Independent Handover
MMS	Multimedia Messaging Service
MOPED	Mobile Grouped Device

MPDU	MAC Protocol Data Unit
MPR	Multipoint Relay
MR	Mobile Router
MTM	Medium Time Metric
MTU	Maximum Transmission Unit
NAD	Network Access Device
NAPT	Network Address Port Translation
NAT	Network Address Translator
NEMO	Network Mobility
NEXWAY	Network of Excellence in Wireless Applications and Technology
NHDP	Neighborhood Discovery Protocol
OLSR	Optimized Link State Routing Protocol
PAC	Proximity Authenticated Channel
PACWOMAN	Power Aware Communications for Wireless Optimised Personal Area Network
PAN	Personal Area Network
PC	Personal Computer
PDA	Personal Digital Assistant
PDE	Personal Distributed Environment
PFS	Prioritized Flooding with Self-Pruning
PGP	Pretty Good Privacy
PID	Personal ID
PKI	Public Key Infrastructure
PMH	Personal Mobile Hub
PNDB	Personal Network Database
PN-F	PN Federation
PNNT	Personal Node Neighbor Table
PNP2008	Personal Network Pilot 2008
PN	Personal Network
P-PAN	Private Personal Area Network
PRNET	Packet Radio Network
QoS	Quality of Service
RAD	Random Assessment Delay
RFF	Radio Frequency Fingerprinting
ROAM	Robust Overlay Architecture for Mobility
RSSI	Received Signal Strength Indication
RTS	Request to Send
RVS	Rendezvous Server
SACK	Selective Acknowledgment
SBA	Scalable Broadcasting Algorithm
SCP	Secure copy
SHAMAN	Security for Heterogeneous Access in Mobile Applications and Networks

SHR	Synchronization Header
SIM	Subscriber Identity Module
SMN	Service Management Node
SMS	Short Message Service
SNR	Signal to Noise Ratio
SPI	Security Parameter Index
SSA	Signal Stability-Based Adaptive Routing
SSH	Secure Shell
STT	Single Trip Transmission Time
STUN	Simple Traversal of UDP Through NATs
TCP	Transmission Control Protocol
TEP	Tunnel Endpoint
TURN	Traversal using Relay NAT
UCL	Universal Convergence Layer
UDP	User Datagram Protocol
UIA	User Information Architecture
UIP	Unmanaged Internet Protocol
UMTS	Universal Mobile Telecommunications System
UPN	Universal Personal Networking
USB	Universal Serial Bus
VoIP	Voice over IP
WAN	Wide Area Network
WCETT	Weighted Cumulative ETT
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WPA	Wireless Protect Access
WSI	Wireless Strategic Initiative
WWI	Wireless World Initiative
WWRF	Wireless World Research Forum
ZRP	Zone Routing Protocol

Appendix B

Accompanying Material

Accompanying this thesis, some electronic material has been made available. This material can be downloaded from the following locations:

`http://www.wmc.ewi.tudelft.nl/~martin/thesis/`
`http://www.exmandato.se/~martin/thesis/`

The content of this material is as follows:

`README`

This file contains instructions on how to use this material.

`thesis.pdf`

This is the thesis itself in Adobe Acrobat PDF format.

`flooding-results.xls`

This file, in Microsoft Excel format, contains the raw simulation and measurement results as shown in the thesis in Chapter 5 on Cluster-wide broadcasting (flooding).

`lqa-results.xls`

This file, in Microsoft Excel format, contains the raw measurement results as shown in the thesis in Chapter 6 on intra-Cluster routing and link quality assessment (LQA).

`software/`

> This folder contains third party software that is required to install and run the simulations and the prototype.

`simulation/`

> This folder contains a patch for ns-2.27 that introduces flooding capabilities for all flooding protocols we simulated in the thesis. The folder also contains scripts for reproducing the simulations themselves.

`pnproto/`

> This folder contains the source code for the PN prototype that was introduced in the thesis. This includes the source code for ppand and gwd as well as patches for madwifi-0.9.3.2 and olsrd-0.4.10. To try the code, you need a Linux system with a 2.6 kernel.

Appendix C

Supporting Projects

In addition to Delft University of Technology, the work in this thesis has been funded by the following research projects. Below, we present each of them together with their official presentations.

IST NEXWAY, IST MAGNET, IST MAGNET Beyond are all funded under the Sixth Framework Programme (FP6). 'Framework programmes' (FPs) have been the main financial tools through which the European Union supports research and development activities covering almost all scientific disciplines. FPs are proposed by the European Commission and adopted by the Council and the European Parliament following a co-decision procedure.

IST NEXWAY

Network of Excellence in Wireless Applications and Technology (NEXWAY) is a Thematic Network in Wireless Communications, with duration of 18 months. It seeks to provide a proof of concept for a new type of Network of Excellence in view of the specified goals and priorities of the Sixth Framework Program.

The purpose of NEXWAY is to build a strong and open team based upon a pool of Academic and Independent R&D Organizations with international reputation in the field of Wireless Communications in order to serve the European Society and Industry.

Project number: FP6-IST-2001-37944
This project concluded in 2004.
`http://www.telecom.ece.ntua.gr/nexway/`

IST MAGNET

MAGNET — My personal Adaptive Global NET — is a worldwide R&D project within Mobile and Wireless Systems and Platforms Beyond 3G. MAGNET will introduce new technologies, systems, and applications that are at the same time user-centric and secure. MAGNET will develop user-centric business model concepts for secure Personal Networks in multi-network, multi-device, and multi-user environments. MAGNET has 37 partners from 17 countries, -highly acknowledged Industrial Partners, Universities, and Research Centres.

Project number: FP6-IST-IP-507102
This project concluded in 2005.
`http://www.ist-magnet.org/`

IST MAGNET Beyond

MAGNET Beyond is a continuation of the MAGNET project. MAGNET Beyond is a worldwide R&D project within Mobile and Wireless Systems and Platforms Beyond 3G. MAGNET Beyond will introduce new technologies, systems, and applications that are at the same time user-centric and secure. MAGNET Beyond will develop user-centric business model concepts for secure Personal Networks in multi-network, multi-device, and multi-user environments. MAGNET Beyond has 32 partners from 15 countries, among these highly influential Industrial Partners, Universities, Research Centres, and SMEs.

Project number: FP6-IST-IP-027396
This project concludes in 2008.
`http://www.ist-magnet.org/`

Freeband PNP2008

(this presentation is shortened)

The PNP2008 project is part of the Freeband Communication programme, which aims at the generation of public knowledge in advanced telecommunication (technology and applications). Freeband is based on the vision of 4G networks and services. It specifically aims at establishing, maintaining and reinforcing the Dutch knowledge position at the international forefront of scientific and technological developments, addressing the most urgent needs for research and novel applications in the present unfolding of new technology. Freeband comprises more than 25 organisations, including all-important technology providers and many representative end-user organisations. The Dutch Ministry of Economic Affairs is co-funding this programme as part of the BSIK plan.

The goal of the PNP2008 project is to develop and demonstrate the novel concept of the Personal Network (PN), which is a distributed personal environment consisting of clusters of geographically dispersed devices that dynamically changes according to the context and needs of the user. Preparing and running a real-life pilot once a year, starting from the first year of the project, will provide a unique insight and feedback in the technical, business and user-related issues associated with the introduction of PN. A distinctive element of this project is the investigation, development and demonstration of the concept of a Personal Network Gateway, an important enabling factor for the incorporation of the Personal Area Network into a fully functional PN. As important will be a Mobility Provider platform that provides an operational environment to manage user, service and network related issues. The main results foreseen by the project are in the field of network architectures and protocols, security, biometric authentication, mobility management and user aspects. Developing an automated and context sensitive concept is a challenging research task, but it has a strong industrial potential, since it would bring in the possibility to build a whole new class of applications, services and devices.

This project concludes in 2008.
`http://pnp2008.freeband.nl/`

Bibliography

[1] 3Com OfficeConnect Wireless 108Mbps 11g XJACK PC Card, `http://www.3com.com/prod/en\_EU\_EMEA/detail.jsp?tab=features\&sku=3CRXJK10075`, Product Specification, Accessed in March 2008.

[2] 3rd Generation Partnership Project, *Service requirements for Personal Network Management (PNM) - Stage 1*, Technical Specification, 3GPP TS 22.259 V8.3.0 (2006-06), March 2007.

[3] 3rd Generation Partnership Project, *Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface*, Technical Specification, 3GPP, TS 51.011, Version 4.15., June 16, 2005.

[4] IST-2001-37385 6HOP, *D2.3 - Analyses of Measurements and Simulations in Multi-hop Ad Hoc Environment*, July 8, 2004.

[5] Mehran Abolhasan, Tadeusz Wysocki and Eryk Dutkiewicz, A Review of Routing Protocols for Mobile Ad Hoc Networks, *Ad Hoc Networks*, Volume: 2, Issue: 1, Pages: 1 – 22, January 2004.

[6] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, Pete Steggles, Towards a Better Understanding of Context and Context-Awareness, In *the First International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, Karlsruhe, Germany, September 1999.

[7] Cedric Adjih, Philippe Jacquet, Laurent Viennot, Computing Connected Dominated Sets with Multipoint Relays, *Ad Hoc & Sensor Networks*, Volume: 1, Issue: 1-2, Pages: 27 – 39, OCP Science, May 2005.

[8] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, Robert Morris, Link-level Measurements from an 802.11b Mesh Network, In *ACM SIGCOMM Conference 2004*, Portland (OR), USA, August 30 - September 3, 2004.

[9] Werner Almesberger, Linux Network Traffic Control - Implementation Overview, In *the 5th Annual Linux Expo*, Raleigh (NC), USA, May 1999.

[10] Ambient Networks (AN), `http://www.ambient-networks.org/`.

[11] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, Enrico Gregori, Wi-Fi in Ad Hoc Mode: A Measurement Study, In *the Second IEEE Annual Conference on Pervasive Computing and Communications (PerCom'04)*, Orlando (FL), USA, March 14 - 17, 2004.

[12] Furquan Ansari, Ajay Sathyanath, STEM: Seamless Transport Endpoint Mobility, *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, Volume: 11, Issue: 2, Pages: 1 – 13, April 2007.

[13] Anish Arora, Rajiv Ramnath, Prasun Sinha, Emre Ertin, Sandip Bapat, Vinayak Naik, Vinod Kulathumani, Hongwei Zhang, Mukundan Sridharan, Santosh Kumar, Hui Cao, Nick Seddon, Chris Anderson, Ted Herman, Chen Zhang, Nishank Trivedi, Mohamed Gouda, Young-ri Choi, Mikhail Nesterenko, Romil Shah, Sandeep Kulkarni, Mahesh Aramugam, Limin Wang, David Culler, Prabal Dutta, Cory Sharp, Gilman Tolle, Mike Grimmer, Bill Ferriera, Ken Parker, Project ExScal (Short Abstract), In *the International Conference on Distributed Computing in Sensor Systems (DOCSS'05)*, Marina del Rey (CA), USA, June 30 - July 1, 2005.

[14] Arvind, Jamey Hicks, A Mobile Phone Ecosystem: MIT and Nokia's Joint Research Venture, *IEEE Intelligent Systems*, Volume: 21, Issue: 5, Pages: 78 – 79, September/October 2006.

[15] Freeband Awareness, `http://awareness.freeband.nl/`.

[16] Baruch Awerbuch, David Holmer, Herbert Rubens. High Throughput Route Selection in Multi-Rate Ad Hoc Wireless Networks. Technical Report Technical Report, Version 2, John Hopkins University, Baltimore (MD), USA, March 12, 2003.

[17] Hari Balakrishnan, Srinivasan Seshan, Randy H. Katz, Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks, *ACM Wireless Networks*, Volume: 1, Issue: 4, Pages: 469 – 481, December 1995.

[18] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, Randy H. Katz, A Comparison of Mechanisms for Improving TCP Performance over Wireless Links, *IEEE/ACM Transactions on Networking (TON)*, Volume: 5, Issue: 6, Pages: 756 – 769, December 1997.

[19] Dirk Balfanz, Glenn Durfee, D. K. Smetters, Rebecca E. Grinter, In Search of Usable Security: Five Lessons from the Field, *IEEE Security and Privacy*, Volume: 2, Issue: 5, Pages: 19 – 24, September-October 2004.

[20] Martin Bauer, Rasmus L. Olsen, Martin Jacobsson, Luis Sanchez, Jorge Lanza, Mohammed Imine, Neeli Prasad, Context Management Framework for MAGNET Beyond, In *the Open International Workshop on Capturing Context and Context Aware Systems and Platforms*, Myconos, Greece, June 8, 2006.

[21] John C. Bicket. Bit-rate Selection in Wireless Networks. Master's Thesis, Massachusetts Institute of Technology, USA, February 2005.

[22] John C. Bicket, Daniel Aguayo, Sanjit Biswas, Robert Morris, Architecture and Evaluation of an Unplanned 802.11b Mesh Network, In *the Annual International Conference on Mobile Computing and Networking (MobiCom'05)*, Cologne, Germany, August 28 - September 2, 2005.

[23] Bluetooth Tracking, `http://www.bluetoothtracking.org/`.

[24] Lawrence S. Brakmo, Larry L. Peterson, TCP Vegas: End to End Congestion Avoidance on a Global Internet, *IEEE Journal on Selected Areas in Communications*, Volume: 1, Issue: 8, Pages: 1465 – 1480, 1995.

[25] Katrin Braun, Joachim Grollman, Michael Horn, Hartmut Raffler, Wolfgang Thulke, Walter Weigel, Universal Personal Networking, In *the Second International Conference on Universal Personal Communications (ICUPC'93)*, Ottawa (ON), Canada, October 12-15, 1993.

[26] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva, A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols, In *the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98)*, Dallas (TX), USA, October 25-30, 1998.

[27] Pat Calhoun, Bob O'Hara, *802.11r Strengthens Wireless Voice*, `http://www.networkworld.com/news/tech/2005/082205techupdate.html`, Network World, August 22, 2005.

[28] Casey Carter, Robin Kravets, Jean Thourrilhes, Contact Networking: A Localized Mobility System, In *the First International Conference on Mobile Systems, Applications, and Services (MobiSys'03)*, San Fransisco (CA), USA, May 5-8, 2003.

[29] David Cavin, Yoav Sasson, André Schiper, On the accuracy of MANET Simulators, In *the Second ACM International Workshop on Principles of Mobile Computing (POMC'02)*, Toulouse, France, October 2002.

[30] Novi I. CempakaWangi, R. Venkatesha Prasad, Martin Jacobsson, Ignas G.M.M. Niemegeers, Address Autoconfiguration in Wireless Ad Hoc Networks: Protocols and Techniques, *IEEE Wireless Communications Magazine*, Volume: 15, Issue: 1, February 2008.

[31] Ian D. Chakeres, Elizabeth M. Belding-Royer, The Utility of Hello Messages for Determining Link Connectivity, In *the 5th International Symposium on Wireless Personal Multimedia Communications*, Honolulu (HI), USA, October 27-30, 2002.

[32] Ian D. Chakeres, Charles E. Perkins, Dynamic MANET On-demand (DYMO) Routing, IETF Internet-Draft (Work in Progress), draft-ietf-manet-dymo-13, April 9, 2008.

[33] Guillaume Chelius, Eric Fleury, Ananas: A Local Area Ad hoc Network Architectural Scheme, In *the 4th International Workshop on Mobile and Wireless Communications Network (MWCN'02)*, Stockholm, Sweden, September 9-11, 2002.

[34] Xiang Chen, Hongqiang Zhai, Jianfeng Wang, Yuguang Fang, TCP Performance over Mobile Ad Hoc Networks, *Canadian Journal of Electrical and Computer Engineering (CJECE), Special Issue on Advances in Wireless Communications and Networking*, Volume: 29, Issue: 1/2, Pages: 129 – 134, January/April 2004.

[35] Kwan-Wu Chin, John Judge, Aidan Williams, Roger Kermode, Implementation Experience with MANET Routing Protocols, *ACM SIGCOMM Computer Communication Review*, Volume: 32, Issue: 5, Pages: 49 – 59, November 2002.

[36] JinHyeock Choi, Greg Daley, Goals of Detecting Network Attachment in IPv6, IETF RFC 4135, August 2005.

[37] Thomas H. Clausen, Philippe Jacquet, Optimized Link State Routing Protocol (OLSR), IETF RFC 3626, October 2003.

[38] Thomas H. Clausen, Christopher M. Dearlove, Justin W. Dean, Cedric Adjih, Generalized MANET Packet/Message Format, IETF Internet-Draft (Work in Progress), draft-ietf-manet-packetbb-12, March 7, 2008.

[39] Thomas H. Clausen, Christopher M. Dearlove, Justin W. Dean, MANET Neighborhood Discovery Protocol (NHDP), IETF Internet-Draft (Work in Progress), draft-ietf-manet-nhdp-06, March 10, 2008.

[40] Thomas H. Clausen, Christopher M. Dearlove, Philippe Jacquet, The Optimized Link State Routing Protocol version 2, IETF Internet-Draft (Work in Progress), draft-ietf-manet-olsrv2-05, February 25, 2008.

[41] Anneli Dahlström, Fredrik Heintz, Martin Jacobsson, Johan Thapper, Martin Öberg, The NOAI Team Description, *RoboCup-2000: Robot Soccer World Cup IV*, Lecture Notes in Computer Science, Volume: 2019, Edited by Peter Stone et al., Pages: 413–416, Springer Verlag, January 2001.

[42] Fei Dai, Jie Wu, An extended localized algorithms for connected dominating set formation in ad hoc wireless networks, *IEEE Transactions on Parallel and Distributed Systems*, Volume: 15, Issue: 10, Pages: 908 – 920, October 2004.

[43] Philippe Debaty, Deborah Caswell, Uniform Web Presence Architecture for People, Places, and Things, *IEEE Personal Communications*, Volume: 8, Issue: 4, Pages: 46 – 51, August 2001.

[44] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, Robert Morris. Effects of Loss Rate on Ad Hoc Wireless Routing. Technical Report Technical Report MIT-LCS-TR-836, Massachusetts Institute of Technology, USA, March 8, 2002.

[45] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, Robert Morris, Performance of Multihop Wireless Networks: Shortest Path is not Enough, *ACM SIGCOMM Computer Communications Review*, Volume: 33, Issue: 1, Pages: 83 – 88, January 2003.

[46] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, Robert Morris, Performance of Multihop Wireless Networks: Shortest Path is Not Enough, *ACM SIGCOMM Computer Communications Review*, Volume: 33, Issue: 1, January 2003.

[47] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, Robert Morris, A High-Throughput Path Metric for Multi-Hop Wireless Routing, In *the Annual International Conference on Mobile Computing and Networking (MobiCom'03)*, San Diego (CA), USA, September 14-19, 2003.

[48] Stephen E. Deering, Robert M. Hinden, Internet Protocol, Version 6 (IPv6) Specification, IETF RFC 2460, December 1998.

[49] Vijay Devarapalli, Ryuji Wakikawa, Alexandru Petrescu, Pascal Thubert, Network Mobility (NEMO) Basic Support Protocol, IETF RFC 3963, January 2005.

[50] Wenqing Ding, Abbas Jamalipour, A New Explicit Loss Notification with Acknowledgment for Wireless TCP, In *the 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'01)*, San Diego (CA), USA, September 30 - October 3, 2001.

[51] Andy Dornan, *The Essential Guide to Wireless Communications Applications*, Prentice-Hall, 2001.

[52] Paul Dourish, Rebecca E. Grinter, Jessica Delgado de la Flor, Melissa Joseph, Security in the Wild: User Strategies for Managing Security as an Everyday Practical Problem, *Personal and Ubiquitous Computing*, Volume: 8, Issue: 6, Pages: 391 – 401, Springer, November 2004.

[53] Richard Draves, Jitendra Padhye, Brian Zill, Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks, In *the Annual International Conference on Mobile Computing and Networking (MobiCom'04)*, Philadelphia, USA, September 26 - October 1, 2004.

[54] Ralph Droms, Dynamic Host Configuration Protocol, IETF RFC 2131, March 1997.

[55] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, Satish K. Tripathi, Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks, *IEEE Personal Communications*, Volume: 4, Issue: 1, Pages: 36 – 45, February 1997.

[56] John Dunlop, R.C. Atkinson, James M. Irvine, D. Pearce, A Personal Distributed Environment for Future Mobile Systems, In *the 12th IST Mobile & Wireless Communication Summit*, Aveiro, Portugal, June 15-18, 2003.

[57] John Dunlop. The Concept of a Personal Distributed Environment for Wireless Service Delivery. Technical report, NEXWAY White Paper, June 2004.

[58] Ethertap, `http://vtun.sourceforge.net/tun/`.

[59] Bryan Ford, Unmanaged Internet Protocol: Taming the Edge Network Management Crisis, In *the Second Workshop on Hot Topics in Networks (HotNets-II)*, Cambridge (MA), USA, November 20-21, 2003.

[60] Zhenghua Fu, Petros Zerfos, Haiyun Luo, Songwu Lu, Lixia Zhang, Mario Gerla, The Impact of Multihop Wireless Channel on TCP Throughput and Loss, In *the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, San Francisco (CA), USA, March 30 - April 3, 2003.

[61] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, S. Wicker. Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks. Technical Report UCLA/CSD-TR 02-0013, UCLA Computer Science, 2002.

[62] Simson Garfinkel, *Database Nation: The Death of Privacy in the 21st Century*, O'Reilly and Associates, January 2000.

[63] Christian Gehrmann, Kaisa Nyberg, Chris Mitchell, The personal CA - PKI for a Personal Area Network, In *the 11th IST Mobile & Wireless Telecommunications Summit*, Thessaloniki, Greece, June 16-19, 2002.

[64] Christian Gehrmann, Thomas Kuhn, Kaisa Nyberg, Peter Windirsch, Trust model, communication and configuration security for Personal Area Networks, In *the 11th IST Mobile & Wireless Telecommunications Summit*, Thessaloniki, Greece, June 16-19, 2002.

[65] Majid Ghader, Rasmus L. Olsen, Marc Girod Genet, Rahim Tafazolli, Service Management Platform for Personal Networks, In *the 14th IST Mobile & Wireless Communications Summit*, Dresden, Germany, June 19-23, 2005.

[66] Majid Ghader, Rasmus L. Olsen, R. Venkatesha Prasad, Martin Jacobsson, Luis Sanchez, Jorge Lanza, Wassef Louati, Marc Girod Genet, Djamal Zeghlache, Rahim Tafazolli, Service Discovery in Personal Networks; Design, Implementation and Analysis, In *the 15th IST Mobile & Wireless Communications Summit*, Myconos, Greece, June 4-8, 2006.

[67] Saikat Guha, Neil Daswani, Ravi Jain, An Experimental Study of the Skype Peer-to-Peer VoIP System, In *the 5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, Santa Barbara (CA), USA, February 27-28, 2006.

[68] Zygmunt J. Haas, A Routing Protocol for the Reconfigurable Wireless Networks, In *the 6th International Conference on Universal Personal Communications (ICUPC'97)*, San Diego (CA), USA, October 12-16, 1997.

[69] Zygmunt J. Haas, Marc R. Pearlman, The Performance of Query Control Schemes for the Zone Routing Protocol, *IEEE/ACM Transactions on Networking*, Volume: 9, Issue: 4, Pages: 427 – 438, August 2001.

[70] Jeyanthi Hall, Michel Barbeau, Evangelos Kranakis, Detection of Transient in Radio Frequency Fingerprinting using Signal Phase, In *the Wireless and Optical Communications Conference (WOC'03)*, Banff (AB), Canada, July 14-16, 2003.

[71] Kevin He, Kernel Korner - Why and How to Use Netlink Socket, *Linux Journal*, `http://www.linuxjournal.com/article/7356`, January 2005.

[72] Geert J. Heijenk, Fei Liu, Interference-Based Routing in Multi-Hop Wireless Infrastructures, *Computer Communications*, Volume: 29, Issue: 13-14, Pages: 2693 – 2701, August 2006.

[73] Robert M. Hinden, Stephen E. Deering, IP Version 6 Addressing Architecture, IETF RFC 4291, February 2006.

[74] Jeroen Hoebeke, Ingrid Moerman, Bart Dhoedt, Piet Demeester, Adaptive Multi-Mode Routing in Ad Hoc Networks, In *FIP TC6 9th International Conference on Personal Wireless Communications (PWC'04)*, Delft, The Netherlands, September 21-23, 2004.

[75] Jeroen Hoebeke, Gerry Holderbeke, Ingrid Moerman, Martin Jacobsson, R. Venkatesha Prasad, Novi I. Cempaka Wangi, Ignas G.M.M. Niemegeers, Sonia M. Heemstra de Groot, Personal Network Federations, In *the 15th IST Mobile & Wireless Communications Summit*, Myconos, Greece, June 4-8, 2006.

[76] Jeroen Hoebeke, Gerry Holderbeke, Ingrid Moerman, Wajdi Louati, Wassef Louati, Marc Girod Genet, Djamal Zeghlache, Luis Sanchez, Jorge Lanza, Mikko Alutoin, Kimmo Ahola, Sami Lehtonen, Jordi Jaen Pallares, Personal Networks: From Concept to a Demonstrator, In *the 15th IST Mobile & Wireless Communications Summit*, Myconos, Greece, June 4-8, 2006.

[77] Jeroen Hoebeke. Adaptive Ad Hoc Routing and Its Application to Virtual Private Ad Hoc Networks. PhD Thesis, Universiteit Gent, Gent, Belgium, November 2007.

[78] Jonathan W. Hui, David Culler, The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale, In *the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore (MD), USA, November 3-5, 2004.

[79] Dirk Husemann, Chandra Narayanaswa, Michael Nidd, Personal Mobile Hub, In *the Eighth IEEE International Symposium on Wearable Computers (ISWC'04)*, Arlington (VA), USA, October 31 - November 3, 2004.

[80] Ari Huttunen, Brian Swander, Victor Volpe, Larry DiBurro, Markus Stenberg, UDP Encapsulation of IPsec ESP Packets, IETF RFC 3948, January 2005.

[81] Malohat Ibrohimovna Kamilova, Sonia M. Heemstra de Groot, Proxy-Based Fednets for Sharing Personal Services in Distributed Environments, In *the Fourth International Conference on Wireless and Mobile Communications (ICWMC'08) (to appear)*, Athens, Greece, July 27 - August 1, 2008.

[82] IEEE 802.11, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, ANSI/IEEE Std 802.11, ISO/IEC 8802-11: 1999, 1999 edition, 1999.

[83] IEEE 802.11i, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Std. 802.11i-2004, 2004 edition, July 2004.

[84] IEEE 802.15.1, *Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs(TM))*, ANSI/IEEE Std 802.15.1, 2005 edition, June 2005.

[85] IEEE 802.15.3, *Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN)*, IEEE Std 802.15.3, 2003 edition, September 2003.

[86] IEEE 802.15.3b, *Part 15.3b: Wireless Medium Access Control(MAC) and Physical Layer (PHY)Specifications for High Rate Wireless Personal Area Networks (WPANs) Amendment 1 : MAC Sublayer*, IEEE Std. 802.15.3b, 2005 edition, May 2006.

[87] IEEE 802.15.4, *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4, 2003 edition, October 2003.

[88] IEEE 802.16, *Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Std 802.16, 2004 edition, October 2004.

[89] IEEE 802.16e, *Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1*, IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005, February 2006.

[90] IEEE 802.21, `http://www.ieee802.org/21/`.

[91] IEEE, *Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority*, `http://standards.ieee.org/regauth/oui/tutorials/EUI64.html`, March 1997.

[92] Infrared Data Association, `http://www.irda.org/`.

[93] Internet World Stats, `http://www.internetworldstats.com/`, Miniwatts Marketing Group, Accessed in April 2008.

[94] Information Society Technologies Advisory Group (ISTAG). Scenarios for Ambient Intelligence in 2010. Technical report, `ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf`, February 2001.

[95] Martin Jacobsson. Resource Management in Differentiated Services - A Prototype Implementation. M.Sc. Thesis, Computer Science/TSS, University of Twente, The Netherlands, June 2001.

[96] Martin Jacobsson, Jeroen Hoebeke, Sonia M. Heemstra de Groot, Anthony Lo, Ingrid Moerman, Ignas G.M.M. Niemegeers, A Network Layer Architecture for Personal Networks, In *the First MAGNET Workshop*, Shanghai, China, October 17, 2004.

[97] Martin Jacobsson, Ignas G.M.M. Niemegeers, Privacy and Anonymity in Personal Networks, In *the 2nd International Workshop on Pervasive Computing and Communication Security (PerSec'05)*, Kauai Island, Hawaii, USA, March 8, 2005.

[98] Martin Jacobsson, Jeroen Hoebeke, Sonia M. Heemstra de Groot, Anthony Lo, Ingrid Moerman, Ignas G.M.M. Niemegeers, Luis Muñoz, Mikko Alutoin, Wajdi Louati, Djamal Zeghlache, A Network Architecture for Personal Networks, In *the 14th IST Mobile & Wireless Communications Summit*, Dresden, Germany, June 19-23, 2005.

[99] Martin Jacobsson, Cheng Guo, Ignas G.M.M. Niemegeers, A Flooding Protocol for MANETs with Self-Pruning and Prioritized Retransmissions, In *the International Workshop on Localized Communication and Topology Protocols for Ad hoc Networks (LOCAN'05)*, Washington DC, USA, November 7-10, 2005.

[100] Martin Jacobsson, R. Venkatesha Prasad, Weidong Lu, Ignas G.M.M. Niemegeers, Foreign Communication in Personal Networks, In *the Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'06)*, Lipari, Italy, June 14-17, 2006.

[101] Assed Jehangir, Sonia M. Heemstra de Groot, Securing Inter-Cluster Communication in Personal Networks, In *the Second International Workshop on Personalized Networks (Pernets'07)*, Philadelphia (PA), USA, August 10 2007.

[102] David B. Johnson, David A. Maltz, Dynamic Source Routing in Ad-Hoc Wireless Network, In *ACM SIGCOMM Conference 1996*, Stanford University (CA), USA, August 28-30, 1996.

[103] David B. Johnson, Charles E. Perkins, Jari Arkko, Mobility Support in IPv6, IETF RFC 3775, June 2004.

[104] David B. Johnson, David A. Maltz, Yih-Chun Hu, The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks, IETF RFC 4728, February 2007.

[105] Petri Jokela, Robert Moskowitz, Pekka Nikander, Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP), IETF RFC 5202, April 2008.

[106] Glenn Judd, Xiaohui Wang, Peter Steenkiste, Low-Overhead Channel-Aware Rate Adaptation, In *the Thirteenth Annual International Conference on Mobile Computing and Networking (MobiCom'07)*, Montreal (QC), Canada, September 9-14, 2007.

[107] Frans Kaashoek, Robert Morris, User-Relative Names for Globally Connected Personal Devices, In *the 5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, Santa Barbara (CA), USA, February 27-28, 2006.

[108] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, R. C. Kunzelman, Advances in Packet Radio Technology, *The Proceedings of the IEEE*, Volume: 66, Issue: 11, Pages: 1468 – 1496, November 1978.

[109] Stephen Kent, IP Encapsulating Security Payload (ESP), IETF RFC 4303, December 2005.

[110] Stephen Kent, Karen Seo, Security Architecture for the Internet Protocol, IETF RFC 4301, December 2005.

[111] Miika Komu, Thomas Henderson, Philip Matthews, Hannes Tschofenig, Ari Keraenen, Jan Melen, Marcelo Bagnulo, Basic HIP Extensions for Traversal of Network Address Translators and Firewalls, IETF Internet-Draft (Work in Progress), draft-ietf-hip-nat-traversal-03, February 25, 2008.

[112] Rajeev Koodli (Ed.), Fast Handovers for Mobile IPv6, IETF RFC 4068, July 2005.

[113] Ernö Kovacs, Martin Bauer, Usman Javaid, Djamal E. Meddour, Context-aware Personal Networks in Beyond 3G Systems, In *the Open International Workshop on Capturing Context and Context Aware Systems and Platforms*, Myconos, Greece, June 8, 2006.

[114] Robin Kravets, Casey Carter, Luiz Magalhães, A Cooperative Approach to User Mobility, *ACM Computer Communications Review*, Volume: 31, Issue: 5, Pages: 57 – 69, October 2001.

[115] Julien Laganier, Teemu Koponen, Lars Eggert, Host Identity Protocol (HIP) Registration Extension, IETF RFC 5203, April 2008.

[116] Julien Laganier, Lars Egger, Host Identity Protocol (HIP) Rendezvous Extension, IETF RFC 5204, April 2008.

[117] Libmcrypt, `http://mcrypt.hellug.gr/lib/index.html`.

[118] The Siemens LifeWorks Concept, `http://www.siemensenterprise.com/attachments/2gip/LifeWorksWhitePaper.pdf`, White Paper, Accessed March 2008.

[119] Hyojun Lim, Chongkwon Kim, Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks, In *the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM'00)*, Boston (MA), USA, August 11, 2000.

[120] The Linux Kernel Archives, `http://www.kernel.org/`.

[121] Justin Lipman, Paul Boustead, Joe Chicharo, Reliable Optimised Flooding in Ad hoc Networks, In *the IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, Shanghai, China, May 31 - June 2, 2004.

[122] Anthony Lo, Weidong Lu, Martin Jacobsson, R. Venkatesha Prasad, Ignas G.M.M. Niemegeers, Personal Networks: An Overlay Network of Wireless Personal Area Networks and 3G Networks, In *the First International Workshop on Personalized Networks (Pernets'06)*, San Jose, USA, July 21, 2006.

[123] Anthony Lo, Weidong Lu, Martin Jacobsson, Venkatesha Prasad, Ignas G.M.M. Niemegeers, Personal Networks - An Architecture for 4G Mobile Communications Networks, *Telektronikk*, , Issue: 1.07, Pages: 45–58, Telenor, April 2007.

[124] Filip Louagie, Luis Muñoz, Sofoklis Kyriazakos, Paving the Way for the Fourth Generation: A New Family of Wireless Personal Area Networks, In *the 12th IST Mobile & Wireless Communications Summit*, Aveiro, Portugal, June 15-18, 2003.

[125] Wajdi Louati, Djamal Zeghlache, Network-based Virtual Personal Overlay Networks using Programmable Virtual Routers, *IEEE Communications Magazine*, Volume: 43, Issue: 8, Pages: 86 – 94, August 2005.

[126] Reiner Ludwig, Randy H. Katz, The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions, *ACM Computer Communications Review*, Volume: 30, Issue: 1, January 2000.

[127] Max do Val Machado, Olga Goussevskaia, Raquel A. F. Mini, Cristiano G. Rezende, Antonio A. F. Loureiro, Geraldo Robson Mateus, José Marcos S. Nogueira, Data Dissemination in Autonomic Wireless Sensor Networks, *IEEE Journal on Selected Areas in Communications*, Volume: 23, Issue: 12, Pages: 2305 – 2319, December 2005.

[128] Joseph Macker (Ed.), Simplified Multicast Forwarding for MANET, IETF Internet-Draft (Work in Progress), draft-ietf-manet-smf-07, February 25, 2008.

[129] Madwifi Driver, `http://www.madwifi.org/`.

[130] IST MAGNET Beyond - My Personal Adaptive Global Net, `http://www.ist-magnet.org/`.

[131] IST-507102 MAGNET/WP1.1/DTU/D1.1.1c/R/PU/001/20.12.2005, *Final User Requirements for the PN Service Architecture*, December 31 2005.

[132] IST-507102 MAGNET/WP1.3/D1.3.1.b/DTU/R/PU/001/1.0, *User Centric Scenarios for PNs of a valid architecture*, December 22 2005.

[133] IST-507102 MAGNET/WP2.1/RWTH/D2.1.2/PU/001/24.10.2005, *Overall Secure PN Architecture*, October 31 2005.

[134] IST-507102 MAGNET/WP2.2/UNIS/D2.2.1/R/PU/001/1.0, *Resource and Service Discovery : PN Solutions*, December 2 2004.

[135] IST-507102 MAGNET/WP2.3/RWTH/D2.3.2/PU/001/19.12.2005, *Ad-hoc self organising and routing architectures (NETWORK layer)*, December 2005.

[136] IST-507102 MAGNET/WP2.4/IMEC/D2.4.1/PU/001/1.0, *Architectures and Protocols for Ad-Hoc Self-configuration, Interworking, Routing and Mobility*, December 2004.

[137] IST-507102 MAGNET/WP3.3/UNIS/D3.3.2b/R/PU/001/1.1, *MAC/RRM Schemes for WPAN (Update D3.3.2a)*, December 2005.

[138] IST-507102 MAGNET/WP4.3/UNIS/D4.3.2/PU/1.00, *Final Version of the Network-Level Security Architecture Specification*, March 3 2005.

[139] IST-027396 MAGNET/B/WP2.3/DUT/D2.3.1/PU/001/12.01.2007, *Specification of PN networking and security components*, January 12, 2007.

[140] IST-027396 MAGNET/B/WP4.1/WMC/D4.1.1/R/RE/001/20.12.-2006, *The Extended Secure Architecture - First Cycle*, December 21, 2006.

[141] IST-027396 MAGNET/B/WP4.2/UNIS/D4.2.1/AR/RE/001/1.0/-20/12/2006, *First Solutions for Implementation of Key Management and Crypto Techniques*, December 31, 2006.

[142] David Maltz, Pravin Bhagwat, Msocks: An Architecture for Transport Layer Mobility, In *the 17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'98)*, San Fransisco, USA, March 29 - April 2, 1998.

[143] The Mobile Ad-hoc Networks (MANET) Charter, `http://www.ietf.org/html.charters/manet-charter.html`.

[144] Petros Maniatis, Mema Roussopoulos, Ed Swierk, Kevin Lai, Guido Appenzeller, Xinhua Zhao, Mary Baker, The Mobile People Architecture, *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, Volume: 3, Issue: 3, Pages: 36 – 42, July 1999.

[145] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, Ren Wang, TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links, In *the 7th Annual International Conference on Mobile computing and Networking (MobiCom'01)*, Rome, Italy, July 16-21, 2001.

[146] Matt Mathis, Jamshid Mahdavi, Sally Floyd, Allyn Romanow, TCP Selective Acknowledgment Options, IETF RFC 2018, October 1996.

[147] Patricia McDermott-Wells, Bluetooth Scatternet Models, *IEEE Potentials*, Volume: 23, Issue: 5, Pages: 36 – 39, December 2004.

[148] Paul Mockapetris, Domain Names - Concepts and Facilities, IETF RFC 1034, Std. 13, November 1987.

[149] Nicolas Montavont, Ryuji Wakikawa, Thierry Ernst, Chan-Wah Ng, Koojana Kuladinithi, Analysis of Multihoming in Mobile IPv6, IETF Internet-Draft (Work in Progress), draft-ietf-monami6-mipv6-analysis-02, May 3, 2008.

[150] Gabriel E. Montenegro, (Ed.), Reverse Tunneling for Mobile IP, revised, IETF RFC 3024, January 2001.

[151] Robert Moskowitz, Pekka Nikander, Petri Jokela, Thomas R. Henderson, Host Identity Protocol, IETF RFC 5201, April 2008.

[152] Moteiv Corporation. *T-mote Sky: Ultra Low Power IEEE 802.15.4 Compliant Wireless Sensor Module (Revision 1.0.2)*, February 2006.

[153] Luis Muñoz, Luis Sanchez, Jorge Lanza, Mikko Alutoin, Kimmo Ahola, Djamal Zeghlache, Marc Girot Genet, Jeroen Hoebeke, Ingrid Moerman, Rasmus L. Olsen, Majid Ghader, Marina Petrova, Martin Jacobsson, A Proposal for Self-Organizing Personal Networks, In *the 15th Wireless World Research Forum (WWRF) Meeting*, Paris, France, December 8-9, 2005.

[154] Homare Murakami, Rasmus L. Olsen, Hans-Peter Schwefel, Ramjee Prasad, Managing Personal Network Specific Addresses in Naming Schemes, In *the Second Interntional MAGNET Workshop*, Aalborg, Denmark, September 17-22, 2005.

[155] Mobile Virtual Centre of Excellence, `http://www.mobilevce.com/`.

[156] Vinayak Naik, Anish Arora, Prasun Sinha, Hongwei Zhang, Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Extreme Scale Wireless Networks of Embedded Devices, *IEEE Transactions on Mobile Computing*, Volume: 6, Issue: 7, Pages: 777 – 789, July 2007.

[157] Sathya Narayanan (Ed.), Detecting Network Attachment in IPv6 Networks (DNAv6), IETF Internet-Draft (Work in Progress), draft-ietf-dna-protocol-07, February 24, 2008.

[158] Thomas Narten, Erik Nordmark, William Allen Simpson, Neighbor Discovery for IP Version 6 (IPv6), IETF RFC 2461, December 1998.

[159] Network Simulator 2, `http://www.isi.edu/nsnam/ns/`.

[160] Chan-Wah Ng, Thierry Ernst, Eun Kyoung Paik, Marcelo Bagnulo, Analysis of Multihoming in Network Mobility Support, IETF RFC 4980, October 2007.

[161] Chan-Wah Ng, Fan Zhao, Masafumi Watari, Pascal Thubert, Network Mobility Route Optimization Solution Space Analysis, IETF RFC 4889, July 2007.

[162] Giao T. Nguyen, Randy H. Katz, Brian Noble, Mahadev Satyanarayanan, A Trace-based Approach for Modeling Wireless Channel Behavior, In *the 28th Conference on Winter Simulation (WSC'96)*, Coronado (CA), USA, December 8-11, 1996.

[163] Ignas G.M.M. Niemegeers, Sonia M. Heemstra de Groot, Research Issues in Ad-Hoc Distributed Personal Networking, *Wireless Personal*

Communications: An International Journal, Volume: 26, Issue: 2-3, Pages: 149 – 167, Kluwer Academic Publishers, August 2003.

[164] Ignas G.M.M. Niemegeers, Sonia M. Heemstra de Groot, FEDNETS: Context-aware Ad-hoc Network Federations, *Wireless Personal Communications: An International Journal*, Volume: 33, Issue: 3-4, Pages: 305 – 318, Springer, June 2005.

[165] Pekka Nikander, Thomas R. Henderson, Christian Vogt, Jari Arkko, End-Host Mobility and Multihoming with the Host Identity Protocol, IETF RFC 5206, April 2008.

[166] Donald A. Norman, *The Psychology of Everyday Things*, Basic Books, New York, 1988.

[167] Katia Obraczka, Kumar Viswanath, Gene Tsudik, Flooding for reliable multicast in multi-hop ad hoc networks, *Wireless Networks*, Volume: 7, Issue: 6, Pages: 627 – 634, Kluwer Academic Publishers, November 2001.

[168] Richard Ogier, MANET Extension of OSPF Using CDS Flooding, In *the 62nd IETF Meeting*, Minneapolis (MN), USA, March 8 2005. `http://www3.ietf.org/proceedings/05mar/slides/ospf-5/sld1.htm`.

[169] The olsr.org OLSR daemon, `http://www.olsr.org/`.

[170] ISO, *Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*, ISO/IEC 7498-1:1994(E), June 1996.

[171] IST PACWOMAN - Power Aware Communications for Wireless Optimised Personal Area Networks, `http://www.imec.be/pacwoman/Welcome.shtml`.

[172] IST-2001-34157 PACWOMAN, *D2.1 - System Requirements and Analysis*, October 8 2002.

[173] Wei Peng, Xi-Cheng Wu, On the Reducation of Broadcast Redundancy in Mobile Ad Hoc Networks, In *the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom'00)*, Boston (MA), USA, August 6-11, 2000.

[174] Wei Peng, Xicheng Lu, AHBP: An Efficient Broadcast Protocol for Mobile Ad Hoc Networks, *Journal of Science and Technology - Beijing, China*, 2002.

[175] Charles E. Perkins, Pravin Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, *ACM SIGCOMM Computer Communication Review*, Volume: 24, Issue: 4, Pages: 234 – 244,, October 1994.

[176] Charles E. Perkins, Elizabeth M. Royer, Ad Hoc On-Demand Distance Vector (AODV) Routing, In *the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, New Orleans (LA), USA, Febuary 25-26, 1999.

[177] Charles E. Perkins, *Ad Hoc Networking*, Addison Wesley, 2001.

[178] Charles E. Perkins, IP Mobility Support for IPv4, IETF RFC 3344, August 2002.

[179] Charles E. Perkins, Elizabeth M. Belding-Royer, Samir R. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561, July 2003.

[180] Freeband Personal Network Pilot 2008 (PNP2008), `http://pnp2008.freeband.nl/`.

[181] Freeband/PNP2008/D1.7v1.0, *Architecture of PNs*, May 2006.

[182] Joseph Polastre, Jason Hill, David Culler, Versatile low power media access for wireless sensor networks, In *Second International Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore (MD), USA, November 3-5, 2004.

[183] Jon Postel, User Datagram Protocol, IETF RFC 768, Std. 6, August 1980.

[184] Jon Postel, Internet Protocol, IETF RFC 791, Std. 5, September 1981.

[185] R. Venkatesha Prasad, Martin Jacobsson, Anthony Lo, Sonia M. Heemstra de Groot, Ignas G.M.M. Niemegeers, Architectures for Communication in Personal Networks, In *the First International Workshop on Personalized Networks (Pernets'06)*, San Jose (CA), USA, July 21, 2006.

[186] R. Venkatesha Prasad, Martin Jacobsson, Sonia M. Heemstra de Groot, Anthony Lo, Ignas G.M.M. Niemegeers, Architectures for Intra-Personal Network Communication, In *the Third ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH'05)*, Cologne, Germany, September 2, 2005.

[187] R. Venkatesha Prasad, Yonghua Li, Martin Jacobsson, Anthony Lo, Ignas G.M.M. Niemegeers, FEW-PNets - A Framework for Emulations of Wireless Personal Networks, In *the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOW-MOM'08) (to appear)*, Newport Beach (CA), USA, June 23-27, 2008.

[188] P2P Universal Computing Consortium (PUCC), `http://www.pucc.jp/`.

[189] Python Programming Language, `http://www.python.org/`.

[190] IOP GenCom QoS for Personal Networks at Home, `http://qos4pn.irctr.tudelft.nl/`.

[191] K. K. Ramakrishnan, Sally Floyd, David L. Black, The Addition of Explicit Congestion Notification (ECN) to IP, IETF RFC 3168, September 2001.

[192] Yakov Rekhter, Robert G. Moskowitz, Daniel Karrenberg, Geert Jan de Groot, Eliot Lear, Address Allocation for Private Internets, IETF RFC 1918, February 1996.

[193] Golden G. Richard III, *Service and Device Discovery - Protocols and Programming*, McGraw-Hill, 2001.

[194] Jonathan Rosenberg, Joel Weinberger, Christian Huitema, Rohan Mahy, STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs), IETF RFC 3489, March 2003.

[195] Jonathan Rosenberg, Rohan Mahy, Christian Huitema, Traversal Using Relay NAT (TURN), IETF Internet-Draft (Work in Progress), draft-rosenberg-midcom-turn-08, September 9, 2005.

[196] Antony Rowstron, Peter Druschel, Pastry: Scalable, Decentralized Object Location and Routing for Large-scale Peer-to-peer Systems, In *the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'01)*, Heidelberg, Germany, November 12-16, 2001.

[197] Joachim Sachs, A Generic Link Layer for Future Generation Wireless Networking, In *the IEEE International Conference on Communications (ICC'03)*, Anchorage (AK), USA, May 11-15, 2003.

[198] Joachim Sachs, H. Wiemann, P. Magnusson, Pontus Wallentin, J. Lundsjö, A generic link layer in a beyond 3G multi-radio access architecture, In *the International Conference on Communications, Circuits and Systems (ICCCAS'04)*, Chengdu, China, June 27-29, 2004.

[199] Luis Sanchez, Jorge Lanza, Luis Muñoz, Julián Pérez Vila, Enabling Secure Communications over Heterogeneous Air Interfaces: Building Private Personal Area Networks, In *the 8th International Symposium on Wireless Personal Multimedia Communications (WPMC'05)*, Aalborg, Denmark, September 18-22, 2005.

[200] Michael Schmidt, Subscriptionless Mobile Networking: Anonymity and Privacy Aspects within Personal Area Networks, In *IEEE Wireless Communications and Networking Conference (WCNC2002)*, Orlando (FL), USA, March 17-21, 2002.

[201] Karsten Schoo, Hokyu Choi, Mohammad Sadegh Fazel, Dirk Dahlhaus, Carlo Mutti, Mauro de Sanctis, P. Balamuralidhar, MC-SS for Personal Area Networks - A Combined PHY and MAC Approach, In *the 14th IST Mobile & Wireless Communications Summit*, Dresden, Germany, June 19-23, 2005.

[202] Karsten Schoo, Franzizkus Bauer, Klaus Strohmenger, Adaptive Modulation and Coding in a PAN Optimized Air Interface Considering Computation Complexity, In *the 15th IST Mobile & Wireless Communications Summit*, Myconos, Greece, June 4-8, 2006.

[203] Henning Schulzrinne, Elin Wedlund, Application-layer mobility using SIP, *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, Volume: 4, Issue: 3, Pages: 47 – 57, July 2000.

[204] Scarlet Schwiderski-Grosche, Allan Tomlinson, David B. Pearce. Towards the Secure Initialisation of a Personal Distributed Environment. Technical Report Technical Report RHUL-MA-2005-09 (`http://www.rhul.ac.uk/mathematics/techreports`), Department of Mathematics, Royal Holloway, University of London, July 20, 2005.

[205] IST-2000-25350 SHAMAN, *D13 - Final technical report - results, specifications and conclusions*, November 30, 2002.

[206] Skype, `http://www.skype.com/`.

[207] Alex C. Snoeren, Hari Balakrishnan, An End-to-End Approach to Host Mobility, In *the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom'00)*, Boston (MA), USA, August 6-11, 2000.

[208] Hesham Soliman, Claude Castelluccia, Karim El Malki, Ludovic Bellier, Hierarchical Mobile IPv6 Mobility Management (HMIPv6), IETF RFC 4140, August 2005.

[209] Pyda Srisuresh, Kjeld B. Egevang, Traditional IP Network Address Translator (Traditional NAT), IETF RFC 3022, January 2001.

[210] Frank Stajano, *Security for Ubiquitous Computing*, John Wiley & Sons, February 2002.

[211] Frank Stajano, Security for Whom? The Shifting Security Assumptions of Pervasive Computing, In *the International Symposium on Software Security (ISSS'02)*, Tokyo, Japan, November 8-10, 2002.

[212] W. Richard Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley Publishing Company, Inc., 1994.

[213] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, Sonesh Surana, Internet Indirection Infrastructure, In *ACM SIGCOMM Conference*, Pittsburgh (PA), USA, August 19-23, 2002.

[214] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, *IEEE/ACM Transactions on Networking (TON)*, Volume: 11, Issue: 1, Pages: 17 – 32, February 2003.

[215] Ivan Stojmenovic, Jie Wu. *Mobile Ad Hoc Networking*, chapter Broadcasting and Activity Scheduling in Ad Hoc Networks, pages 205–230. Wiley-IEEE Press, July 2004.

[216] Thafer Sulaiman, Kumarendra Sivarajah, Hamed S. Al-Raweshidy, Personal Identification (PID) in Personal Area Network (PAN), In *Wireless Personal Multimedia Communications (WPMC'05)*, Aalborg, Denmark, September 18-22, 2005.

[217] Rahim Tafazolli (Editor), *Technologies for the Wireless Future: Wireless World Research Forum (WWRF)*, John Wiley & Sons, October 2004.

[218] Rahim Tafazolli (Editor), *Technologies for the Wireless Future: Wireless World Research Forum (WWRF), Volume 2*, John Wiley & Sons, April 2006.

[219] Ken Tang, Mario Correa, Mario Gerla, Effects of Ad Hoc Layer Medium Access Mechanisms under TCP, *Mobile Networks and Applications*, Volume: 6, Issue: 4, Pages: 317 – 329, August 2001.

[220] Andreas Tarp. Experimental Evaluation of Flooding in ad-hoc Networks. Bachelor's thesis, University of Dusseldorf, Dusseldorf, Germany, December 2004.

[221] Texas Instruments. *CC2420 - 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, March 2007.

[222] TinyOS, `http://www.tinyos.net/`.

[223] C. K. Toh, Associativity-Based Routing for Ad Hoc Mobile Networks, *International Journal on Wireless Personal Communications*, Volume: 4, Issue: 2, Pages: 103 – 139, Kluwer Academic Publishers, March 1997.

[224] Gilman Tolle, David Culler, Design of an Application-Cooperative Management System for Wireless Sensor Networks, In *Second European Workshop on Wireless Sensor Networks (EWSN'05)*, Istanbul, Turkey, Jananuary 31 - February 2 2005.

[225] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, Jang-Ping Sheu, The broadcast storm problem in a mobile ad hoc network, *Wireless Networks*, Volume: 8, Issue: 2/3, Pages: 153 – 167, Kluwer Academic Publishers, May 2002.

[226] Michael Tuexen, Qiaobing Xie, Randall Stewart, Melinda Shore, Lyndon Ong, John Loughney, Maureen Stillman, Requirements for Reliable Server Pooling, IETF RFC 3237, January 2002.

[227] Paul Vixie, Susan Thomson, Yakov Rekhter, Jim Bound, Dynamic Updates in the Domain Name System (DNS UPDATE), IETF RFC 2136, April 1997.

[228] Chieh-Yih Wan, Andrew T. Campbell, Lakshman Krishnamurthy, PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks, In *the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta (GA), USA, September 28, 2002.

[229] Mark Weiser, The Computer for the Twenty-First Century, *Scientific American*, pages 94 – 104, September 1991.

[230] Alma Whitten, J. D. Tygar, Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0, In *the Eight USENIX Security Symposium*, Washington DC, USA, August 23-26, 1999.

[231] Brad Williams, Tracy Camp, Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks, In *the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, Lausanne, Switzerland, June 9-11, 2002.

[232] Wireless Strategic Initiative, *The Book of Visions 2000 - Visions of the Wireless World*, Available at: `http://www.wireless-world-research.org`, November 2000.

[233] Wireless World Research Forum, *The Book of Visions 2001 - Visions of the Wireless World, Version 1.0*, Available at: `http://www.wireless-world-research.org`, December 2001.

[234] The World Factbook, `https://www.cia.gov/cia/publications/factbook/index.html`, Central Intelligence Agency (CIA), Accessed in April 2008.

[235] Lisong Xu, Khaled Harfoush, Injong Rhee, Binary Increase Congestion Control for Fast Long-Distance Networks, In *the 23rd Conference of the IEEE Communications Society (INFOCOM'04)*, Hong Kong, March 7-11, 2004.

[236] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, Lixia Zhang, Security in Mobile Ad Hoc Networks: Challenges and Solutions, *IEEE Wireless Communications*, Volume: 11, Issue: 1, Pages: 38 – 47, February 2004.

[237] Ka-Ping Yee, User Interaction Design for Secure Systems, In *the 4th International Conference on Information and Communications Security (ICICS'02)*, Singapore, December 9-12, 2002.

[238] Victor C. Zandy, Barton P. Miller, Transport Layer Issues: Reliable network connections, In *the 8th annual international conference on Mobile computing and networking (MobiCom'02)*, Atlanta (GA), USA, September 23-28, 2002.

[239] Jinglong Zhou, Anthony Lo, Martin Jacobsson, Ignas G.M.M. Niemegeers, NS-2 Simulation Model for a Novel Beyond 3G Cellular Multihop Network, In *the Workshop on ns-2: the IP network simulator (WNS2)*, Pisa, Italy, October 10, 2006.

[240] Xiaoming Zhou, Martin Jacobsson, Henk Uijterwaal, Piet F.A. van Mieghem, IPv6 Delay and Loss Performance Evolution, *To appear in International Journal of Communication Systems.*

[241] Jinglong Zhou, Martin Jacobsson, Ignas G.M.M. Niemegeers, Cross Layer Design for Enhanced Quality Personal Wireless Networking, In *the Sixth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'07)*, Corfu, Greece, June 13-15, 2007.

[242] Jinglong Zhou, Martin Jacobsson, Ignas Niemegeers, Cross Layer Design for Enhanced Quality Routing in Personal Wireless, In *the Second International Workshop on Personalized Networks (Pernets'07)*, Philadelphia (PA), USA, August 10, 2007.

[243] Jinglong Zhou, Martin Jacobsson, Ertan Onur, Ignas G.M.M. Niemegeers, Factors that Impact Link Quality Estimation in Personal Networks, In *the 8th International Symposium On Computer Networks (ISCN'08) (to appear)*, Istanbul, Turkey, June 18-20, 2008.

[244] Shelley Q. Zhuang, Kevin Lai, Ion Stoica, Randy H. Katz, Scott Shenker, Host Mobility using an Internet Indirection Infrastructure, In *the First International Conference on Mobile Systems, Applications, and Services (ACM/USENIX Mobisys)*, San Fransisco (CA), USA, May 5-8, 2003.

[245] Philip R. Zimmermann, *The official PGP user's guide*, MIT Press, May 1995.

Summary

While a personal area network (PAN) connects a person's devices around him/her, a Personal Network (PN) extends that PAN to other personal devices and services farther away. This extension is physically made via available wired and wireless networks. To be useful, a PN must adapt to changes in the surroundings, be self-configuring, and support as many different types of networks and devices as possible. It must support the person's applications by taking into account the person's context, location, and communication possibilities.

This thesis focuses on the network layer mechanisms of PNs. Given a single user, we propose an architecture for PNs in which there are two types of nodes: personal nodes and foreign nodes. Personal nodes are nodes that belong to the user, while all other nodes are foreign nodes. The PN of a user is the collection of all his/her personal nodes, both remote and within close vicinity. When active personal nodes of a user come together and can communicate with each other without external assistance; they form clusters. A PN is likely to consist of several active clusters, such as a home cluster, an office cluster, a car cluster, etc.

The network level architecture that we propose separates the communication among personal nodes of the same PN from the communication to, from, and among other nodes and devices. To make this happen, each node must know which PN it belongs to and must be able to tell if another neighboring node is a personal node or not. This is achieved by introducing the nodes into the PN by a process we call personalization. Personalization is a prerequisite to any cluster and PN formation, but happens only once when a new node is acquired for the first time by the user. The node then remains a personal node until the user decides otherwise.

Clusters consist solely of active personal nodes that can communicate with each other using their own communication capabilities and without external support. For this, we propose self-organized cluster formation and maintenance mechanisms. Clusters are basically ad hoc networks and may therefore consist of multiple heterogeneous wireless technologies and hence be multi-hop networks. Such networks require special broadcasting and routing protocols. Hence, we investigate cluster-wide broadcasting and how it can be improved. An optimized flooding algorithm for clusters is proposed and its performance compared to other algorithms. We show this by both simulations and measurements in a real wireless network test bed.

Further, we explore how we can improve the ad hoc routing of a cluster. The focus is on how to do link quality assessment (LQA) in an accurate and timely manner. Using available information, including cross-layer information, we can improve the LQA and thereby make the routing more efficient. The benefit is that the routing protocol can make quicker and better routing decisions if it has better link quality information through a better LQA mechanism. To validate our LQA solutions, we developed a prototype with which we tested the gain yielded by different LQA mechanisms. Several options were tested and each of them shows good improvements compared to the standard procedure of using simple hello packets.

To connect the different clusters, we need to use interconnecting structures, such as the Internet. Tunnels between the clusters are established and maintained to allow for communication between clusters. Gateway nodes are personal nodes in the clusters that have connectivity with foreign nodes or the interconnecting structure. It is the responsibility of these nodes to construct the tunnels and interconnect the PN. In order for each cluster to locate the other clusters in the PN, we introduce the concept of PN agent. The role of the PN agent is to coordinate the clusters and keep their locations in a database. In this way, clusters within a PN can easily find each other. Once the tunnels have been established, intra-PN communication can take place. By building and testing a prototype, we show the feasibility of intra-PN communication even with minimal support from the infrastructure.

A PN cannot exist in isolation, but needs to interact with other PNs as well as PN-unaware foreign nodes and other non-IP devices. This, we call foreign communication. It involves both using services from foreign nodes as well as offering services to those nodes. Gateway nodes treat foreign traffic in a different way from intra-PN traffic; non-approved traffic is blocked from entering the PN. Furthermore, gateway nodes bridge the mechanisms used inside the PN with the ones used by the foreign nodes as these mechanisms will be different. Foreign communication is the last topic investigated in this thesis and here, we propose two approaches: one at the network level using network address translators (NATs) and one using service proxies. Solutions for security and mobility aspects are also covered.

The main contribution of this thesis is an architecture for PNs. By working out solutions to the network-related parts of our architecture, we can demonstrate that the architecture is indeed feasible. This, we also prove by successfully making a working prototype that covers all networking aspects of a PN, with the only exception of foreign communication. Our prototype clearly demonstrates that it is possible to build real PNs.

Samenvatting

Een personal area network (PAN) is bedoeld om apparaten die een persoon bezit, met elkaar te verbinden, op voorwaarde dat deze apparaten zich dicht genoeg bij elkaar bevinden om draadloze communicatie over korte afstand toe te staan. Een personal network (PN) daarentegen is een veel breder concept. Een PN is bedoeld om mogelijk te maken dat persoonlijke apparaten en diensten met elkaar verbonden worden en samenwerken, ongeacht waar ze zich bevinden. Een PN kan dus in principe een persoonlijk systeem zijn dat zich geografisch globaal uitstrekt. Dit wordt mogelijk gemaakt door gebruik te maken van zowel draadgebonden als draadloze netwerken, en zowel infrastructuur als ad hoc netwerken. Om van nut te zijn voor zijn gebruiker moet een netwerk zich aanpassen aan veranderingen in de omgeving, het moet zelfconfigurerend zijn en zoveel mogelijk verschillende types netwerken en apparaten ondersteunen. Het moet de toepassingen van de gebruiker ondersteunen door rekening te houden met de context van de gebruiker, de locatie en de communicatie mogelijkheden die beschikbaar zijn.

Dit proefschrift richt zich op de mechanismen binnen de netwerklaag van PNs. Uitgaande van één individuele gebruiker, stellen we een PN architectuur voor met twee typen knooppunten: persoonlijke knooppunten en externe knooppunten. Persoonlijke knooppunten zijn knooppunten die bij de gebruiker horen, alle overige knooppunten zijn externe knooppunten. Het PN van een gebruiker bestaat uit de verzameling van al zijn/haar persoonlijke knooppunten, zowel die op afstand als die dichtbij. Wanneer actieve persoonlijke knooppunten van een gebruiker samenkomen en zonder externe hulp met elkaar kunnen communiceren vormen ze een cluster. Een PN zal doorgaans bestaan uit verschillende actieve clusters, zoals thuisclusters, een kantoorcluster en een cluster in de auto.

De netwerkniveau architectuur die wij voorstellen onderscheidt communicatie tussen persoonlijke knooppunten van het zelfde PN van communicatie naar, vanuit en tussen andere knooppunten en apparaten. Om dit mogelijk te maken moet elk knooppunt weten tot welk PN het behoort en moet het kunnen nagaan of een knooppunt in de buurt een persoonlijk knooppunt is of niet. Dit wordt bereikt door de knooppunten in het PN een proces te laten doorlopen dat we personalisatie noemen. Personalisatie is een voorwaarde voor elke clustervorming en voor de vorming van een PN; het gebeurt echter alleen wanneer een nieuwe knooppunt voor het eerst door de gebruiker wordt aangeschaft. Het knooppunt blijft permanent een persoonlijke knooppunt, tenzij de gebruiker anders beslist.

Clusters bestaan uitsluitend uit actieve persoonlijke knooppunten die met elkaar kunnen communiceren gebruik makend van hun eigen communicatiemogelijkheden en zonder hulp van buitenaf. Hiervoor stellen wij zelfgeorganiseerde cluster vorming- en onderhoudmechanismen voor. Clusters zijn in principe ad hoc netwerken en kunnen daarom bestaan uit meervoudige heterogene draadloze technologieën en zijn dus multi-hop netwerken. Dergelijke netwerken vereisen speciale broadcast en routing protocollen. Om die reden onderzoeken we hoe clusterbrede broadcasting verbeterd kan worden. Een geoptimaliseerd flooding algoritme wordt voorgesteld en zijn prestaties worden vergeleken met andere algoritmen. We laten dit zien door simulaties en metingen in een echte draadloze netwerk test omgeving.

Verder onderzoeken we hoe we de ad hoc routing van een cluster kunnen verbeteren. De nadruk ligt daarbij op hoe de kwaliteit van een draadloze verbinding nauwkeurig en tijdig uitgevoerd kan worden (Link Quality Assessment of LQA). Gebruik makend van de beschikbare informatie, inclusief zogenaamde cross-layer informatie, kunnen we de LQA verbeteren en daarmee de routing efficiënter maken. Het voordeel is dat het routing protocol snellere en betere beslissingen kan nemen indien het over betere informatie over de kwaliteit van de link beschikt. Om onze LQA oplossing te valideren hebben we een prototype ontwikkeld waarmee we verschillende LQA mechanismen getest hebben. Ze vertonen allen sterke verbeteringen ten opzichte van de klassieke LQA methode die gebaseerd is op het gebruik van Hello pakketten.

Voor het verbinden van verschillende clusters maken we gebruik van zogenaamde interconnectiestructuren zoals het Internet. Hiervoor worden tunnels tussen de clusters tot stand gebracht en onderhouden. Gateway knooppunten zijn persoonlijke knooppunten in clusters die kunnen verbonden worden met externe knooppunten of de interconnectiestructuur. Het is de rol van deze knooppunten om tunnels tot stand te brengen en de inter-cluster communicatie van het PN te verzorgen. Om ervoor te zorgen dat de clusters van een PN elkaar kunnen vinden, maken we gebruik van een PN agent. De rol van de PN agent bestaat erin de clusters te coördineren en hun locaties op te slaan in een databank. Zo kunnen clusters elkaar makkelijk vinden. Door een prototype te ontwikkelen tonen we de haalbaarheid van intra-PN communicatie aan, zelfs met minimale steun van infrastructuur.

Een PN kan niet geïsoleerd bestaan, maar moet interactie kunnen hebben met andere PNs en ook met externe knooppunten. We noemen dit externe communicatie. Het gaat zowel om het gebruik van diensten van externe knooppunten als ook het aanbieden van diensten aan deze knooppunten. Gateway knooppunten behandelen extern verkeer anders dan intra-PN verkeer; niet goedgekeurd verkeer wordt geblokkeerd en kan het PN niet binnendringen. Verder slaan de gateways een brug tussen de mechanismen die binnen het PN gebruikt worden en de mechanismen die door externe knooppunten gebruikt worden. Externe communicatie is het laatste onderwerp dat behandeld wordt in het proefschrift. Hier stellen we twee benaderingen voor: één op netwerkniveau gebruik makend van netwerk adres vertalers (Network Address Translators of NATs) en een tweede die gebruik maakt van service proxies. Het proefschrift besteed uiteraard ook aandacht aan de aspecten veiligheid en mobiliteit.

De belangrijkste bijdrage van dit proefschrift is een architectuur voor PNs. Door oplossingen uit te werken voor de netwerk gerelateerde onderdelen van onze architectuur kunnen we aantonen dat de architectuur inderdaad haalbaar is. Dit tonen we tevens aan door een werkend prototype te maken dat alle kenmerkende aspecten van een PN, met uitzondering van externe communicatie, behandelt. Ons prototype laat duidelijk zien dat het mogelijk is om echte PNs te bouwen.

Curriculum Vitae

Personalia

Full Name	Martin Edvard Jakobsson (legal name)
Date of Birth	6 March, 1976
Place of Birth	Boo (Stockholm), Sweden
Sex	Male
Nationality	Sweden

Profile

Martin Jacobsson graduated in Computer Science from University of Linköping, Sweden in 2002. His work experience comprises of software development related to various telecommunications applications. This includes systems for management of Internet provider networks and systems for management of private branch exchange (PBX) systems. The former was made for Telia, a major Internet and telecom provider in Sweden, during 1997 - 2000 and the latter for Philips Business Communications during 2001 and 2002.

In 2003, he joined the Wireless and Mobile Communications group led by professor Niemegeers in Delft University of Technology as a doctoral researcher. There, he participated in several Dutch and European research projects, such as IST NEXWAY, Freeband PNP2008, IST MAGNET, and IST MAGNET Beyond. This thesis is the result of this work.

Work Experience

1993 - 1997	Summer job and temporary part-time at eXmandato, Kalmar, Sweden as a computer consultant. Mainly for Telia, developing network management tools.

Jun. - Aug. 1998 — Full-time summer job at Telia Network Services, Stockholm, Sweden with mainly computer and network management. Operational activities.

Aug. 1998 and
Aug. 1999 — Teaching mathematics to the new undergraduate students at Linköpings Universitet.

Jun. - Aug. 1999 and
Jun. - Sep. 2000 — Full-time summer job at eXmandato, Kalmar, Sweden as a computer consultant. Mainly for Telia developing a QoS measurement tool for IP networks.

Sep. 2001 - Dec. 2002 — Full-time job at Philips Business Communications, Hilversum, The Netherlands as a software engineer. Developed web-based management system for telephone switch equipment.

Jan. 2003 - Present — Scientific Researcher at Delft University of Technology.

Theses, Journal, and Magazine Publications

[41] Anneli Dahlström, Fredrik Heintz, Martin Jacobsson, Johan Thapper, Martin Öberg, The NOAI Team Description, *RoboCup-2000: Robot Soccer World Cup IV*, Lecture Notes in Computer Science, Volume: 2019, Edited by Peter Stone et al., Pages: 413–416, Springer Verlag, January 2001.

[95] Martin Jacobsson. *Resource Management in Differentiated Services - A Prototype Implementation.* M.Sc. Thesis, Computer Science/TSS, University of Twente, The Netherlands, June 2001.

[123] Anthony Lo, Weidong Lu, Martin Jacobsson, Venkatesha Prasad, Ignas G.M.M. Niemegeers, Personal Networks - An Architecture for 4G Mobile Communications Networks, *Telektronikk*, , Issue: 1.07, Pages: 45–58, Telenor, April 2007.

[30] Novi I. CempakaWangi, R. Venkatesha Prasad, Martin Jacobsson, Ignas G.M.M. Niemegeers, Address Autoconfiguration in Wireless Ad Hoc Networks: Protocols and Techniques, *IEEE Wireless Communications Magazine*, Volume: 15, Issue: 1, February 2008.

[240] Xiaoming Zhou, Martin Jacobsson, Henk Uijterwaal, Piet F.A. van Mieghem, IPv6 Delay and Loss Performance Evolution, *To appear in International Journal of Communication Systems.*

Conference and Workshop Publications

[96] Martin Jacobsson, Jeroen Hoebeke, Sonia M. Heemstra de Groot, Anthony Lo, Ingrid Moerman, Ignas G.M.M. Niemegeers, A Network Layer Architecture for Personal Networks, In *the First MAGNET Workshop*, Shanghai, China, October 17, 2004.

[97] Martin Jacobsson, Ignas G.M.M. Niemegeers, Privacy and Anonymity in Personal Networks, In *the 2nd International Workshop on Pervasive Computing and Communication Security (PerSec'05)*, Kauai Island, Hawaii, USA, March 8, 2005.

[98] Martin Jacobsson, Jeroen Hoebeke, Sonia M. Heemstra de Groot, Anthony Lo, Ingrid Moerman, Ignas G.M.M. Niemegeers, Luis Muñoz, Mikko Alutoin, Wajdi Louati, Djamal Zeghlache, A Network Architecture for Personal Networks, In *the 14th IST Mobile & Wireless Communications Summit*, Dresden, Germany, June 19-23, 2005.

[186] R. Venkatesha Prasad, Martin Jacobsson, Sonia M. Heemstra de Groot, Anthony Lo, Ignas G.M.M. Niemegeers, Architectures for Intra-Personal Network Communication, In *the Third ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH'05)*, Cologne, Germany, September 2, 2005.

[99] Martin Jacobsson, Cheng Guo, Ignas G.M.M. Niemegeers, A Flooding Protocol for MANETs with Self-Pruning and Prioritized Retransmissions, In *the International Workshop on Localized Communication and Topology Protocols for Ad hoc Networks (LOCAN'05)*, Washington DC, USA, November 7-10, 2005.

[153] Luis Muñoz, Luis Sanchez, Jorge Lanza, Mikko Alutoin, Kimmo Ahola, Djamal Zeghlache, Marc Girot Genet, Jeroen Hoebeke, Ingrid Moerman, Rasmus L. Olsen, Majid Ghader, Marina Petrova, Martin Jacobsson, A Proposal for Self-Organizing Personal Networks, In *the 15th Wireless World Research Forum (WWRF) Meeting*, Paris, France, December 8-9, 2005.

[75] Jeroen Hoebeke, Gerry Holderbeke, Ingrid Moerman, Martin Jacobsson, R. Venkatesha Prasad, Novi I. Cempaka Wangi, Ignas G.M.M. Niemegeers, Sonia M. Heemstra de Groot, Personal Network Federations, In *the 15th IST Mobile & Wireless Communications Summit*, Myconos, Greece, June 4-8, 2006.

[66] Majid Ghader, Rasmus L. Olsen, R. Venkatesha Prasad, Martin Jacobsson, Luis Sanchez, Jorge Lanza, Wassef Louati, Marc Girod Genet,

Djamal Zeghlache, Rahim Tafazolli, Service Discovery in Personal Networks; Design, Implementation and Analysis, In *the 15th IST Mobile & Wireless Communications Summit*, Myconos, Greece, June 4-8, 2006.

[20] Martin Bauer, Rasmus L. Olsen, Martin Jacobsson, Luis Sanchez, Jorge Lanza, Mohammed Imine, Neeli Prasad, Context Management Framework for MAGNET Beyond, In *the Open International Workshop on Capturing Context and Context Aware Systems and Platforms*, Myconos, Greece, June 8, 2006.

[100] Martin Jacobsson, R. Venkatesha Prasad, Weidong Lu, Ignas G.M.M. Niemegeers, Foreign Communication in Personal Networks, In *the Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'06)*, Lipari, Italy, June 14-17, 2006.

[185] R. Venkatesha Prasad, Martin Jacobsson, Anthony Lo, Sonia M. Heemstra de Groot, Ignas G.M.M. Niemegeers, Architectures for Communication in Personal Networks, In *the First International Workshop on Personalized Networks (Pernets'06)*, San Jose, USA, July 21, 2006.

[122] Anthony Lo, Weidong Lu, Martin Jacobsson, R. Venkatesha Prasad, Ignas G.M.M. Niemegeers, Personal Networks: An Overlay Network of Wireless Personal Area Networks and 3G Networks, In *the First International Workshop on Personalized Networks (Pernets'06)*, San Jose (CA), USA, July 21, 2006.

[239] Jinglong Zhou, Anthony Lo, Martin Jacobsson, Ignas G.M.M. Niemegeers, NS-2 Simulation Model for a Novel Beyond 3G Cellular Multihop Network, In *the Workshop on ns-2: the IP network simulator (WNS2)*, Pisa, Italy, October 10, 2006.

[241] Jinglong Zhou, Martin Jacobsson, Ignas G.M.M. Niemegeers, Cross Layer Design for Enhanced Quality Personal Wireless Networking, In *the Sixth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'07)*, Corfu, Greece, June 13-15, 2007.

[242] Jinglong Zhou, Martin Jacobsson, Ignas Niemegeers, Cross Layer Design for Enhanced Quality Routing in Personal Wireless, In *the Second International Workshop on Personalized Networks (Pernets'07)*, Philadelphia (PA), USA, August 10, 2007.

[243] Jinglong Zhou, Martin Jacobsson, Ertan Onur, Ignas G.M.M. Niemegeers, Factors that Impact Link Quality Estimation in Personal Networks, To Appear in *the 8th International Symposium On Computer Networks (ISCN'08)* , Istanbul, Turkey, June 18-20, 2008.

[187] R. Venkatesha Prasad, Yonghua Li, Martin Jacobsson, Anthony Lo, Ignas G.M.M. Niemegeers, FEW-PNets - A Framework for Emulations of Wireless Personal Networks, To Appear in *the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM'08)*, Newport Beach (CA), USA, June 23-27, 2008.

Other Academic Achievements

- Participated in the following research projects: IST NEXWAY, IST MAGNET, IST MAGNET Beyond, and PNP2008. I was also involved in the defining of new project proposals for European Commission's Framework Programme 6 and 7. In MAGNET Beyond, I was task leader for the tasks related to service control and interworking. In the end of PNP2008, I was work package leader for the work package on PN architectures.

- Supervised or co-supervised the following MSc students in their master thesis work: Yan Gao, Xiang Han, Cheng Guo, and Ting Liu.

- Organized the international workshop on personalized networks (Pernets) 2006 and 2007.

- Reviewer for: the 9th International Conference on Personal Wireless Communications (PWC) 2004, Pernets 2006, Pernets 2007, the Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2008, the International Conference on Computer Communictions and Networks (ICCCN) 2008, the IEEE Global Communications Conference (GLOBECOM) 2008, and IEEE Transactions on Wireless Communication.