Dynamic Auto Configuration and Self-Management of Next Generation Personal Area Networks

João Pedro Carneiro Maia

Dissertation supervised by
Professor Manuel Pereira Ricardo,
of Electrical and Computers Engineering Department
of Faculty of Engineering of the University of Porto

(President of the Jury, Professor José António Ruela)

Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotécnica e Computadores
Rua Roberto Frias, s/n, 4200-465 Porto, Portugal

Porto, March 2008

Locations:

FEUP - Faculty of Engineering of the University of Porto

INESC Porto - Institute for Systems and Computer Engineering of Porto

Supervisors:

Prof. Manuel Pereira Ricardo - FEUP

Eng. Rui Lopes Campos - INESC Porto

Contacts

João Pedro Maia – joao.maia@fe.up.pt

Resumo (Português)

Nas redes de próxima geração prevê-se uma mudança de paradigma de comunicação, nomeadamente do ponto de vista do utilizador. No cenário actual, um utilizador transporta consigo múltiplos dispositivos pessoais que operam de forma isolada. Num cenário futuro esses mesmos dispositivos formarão uma rede pessoal em torno do utilizador, constituindo uma esfera de comunicação que se move com ele e se adapta quer às suas preferências quer ao contexto de comunicação em cada momento.

A heterogeneidade dos dispositivos (*Personal Digital Assistants* (PDAs), Telemóveis, *Laptops*, *Desktops*, Câmara Digitais, Leitores MP3/MP4) e das tecnologias de comunicação que formarão a rede pessoal (Bluetooth, Wi-Fi, Ultra Wide Band (UWB), Ethernet) apresenta-se como um aspecto determinante e com qual é necessário lidar através de novas soluções. As tecnologias actuais apenas permitem a criação de redes pessoais incipientes e focam-se numa única tecnologia de comunicação, por exemplo, Bluetooth ou Wi-Fi (Wireless Fidelity) e não contemplam todos os aspectos referidos anteriormente. Assim, torna-se necessário criar uma nova solução capaz de integrar múltiplas tecnologias de comunicação heterogéneas, considerar a heterogeneidade dos dispositivos que compõem a rede pessoal e a organização automática e dinâmica da rede pessoal atendendo à mobilidade que a caracteriza.

Tendo em conta as limitações tecnológicas actuais Campos e Ricardo propuseram uma nova solução para a autoconfiguração e autogestão de redes pessoais de próxima geração, denominada de *Autoconfiguration and Self-management of Personal Area Networks (ASPAN)*. Neste trabalho desenvolveu-se um protótipo demonstrativo da solução ASPAN, tendo por base as tecnologias Bluetooth e Wi-Fi (ambas sem fios) e Ethernet. A implementação contempla: 1) criação de uma base comum de comunicação independente da tecnologia; 2) autoconfiguração das interfaces de comunicação; 3) eleição de um dispositivo (*master*) responsável pela organização e configuração da rede; 4) descoberta de topologia durante a criação da rede pessoal; 5) protocolo de sinalização utilizado na criação e autogestão da rede pessoal; 6) criação de uma rede IP (*Internet Protocol*) sobre as tecnologias sem fios utilizadas na interligação de dispositivos pessoais;

Abstract

In the next generation networks a change in the communication paradigm, namely from the user point of view, is envisioned. Currently, a user carries multiple personal devices that work independently and in isolation. In a future scenario such devices will instead form a Personal Area Network (PAN) around the user, creating a communicating bubble that moves with him and adapts itself to the user preferences and communication context at each moment in time. The heterogeneity of the personal devices (Personal Digital Assistants (PDAs), Mobile Phones, Laptops, Desktops, Digital Cameras, MP3/MP4 players) and communication technologies that will form a next generation PAN (Bluetooth, Wi-Fi, Ultra Wide Band (UWB), Ethernet) represents a crucial aspect that needs to be tackled with new solutions. Current PAN technologies allow the creation of incipient PANs and only focus on a single technology, such as Bluetooth and Wi-Fi (Wireless Fidelity), and do not consider the entire aspects aforementioned. Thereby, it is needed to come up with a new solution that is able to integrate multiple heterogeneous communication technologies, deal with the heterogeneity of the PAN devices, manage the PAN automatically and dynamically according to the mobility that characterizes it.

Regarding the limitations of current technologies, Campos and Ricardo proposed a new solution targeting the auto configuration and self-management of next generation PANs, called Auto configuration and Self-management of Personal Area Networks (ASPAN). In this work we have developed a demonstrative prototype of the ASPAN solution, taking into account three technologies Bluetooth and Wi-Fi (both wireless), and Ethernet (wired). Our implementation considers: 1) setup of a common communication base independent from the technology; 2) auto configuration of communication interfaces; 3) election of a master device that is in charge of managing and configuring the PAN; 4) PAN topology discovery during bootstrapping; 5) a signaling protocol used to create and manage the PAN; 6) automatic configuration of a IP (Internet Protocol) network within the PAN;

Acknowledgements

This work is based on the ASPAN solution by Campos and Ricardo so my first acknowledgement is for them. Without their guidance this work would not have been possible. I would like especially to thank Eng. Rui Campos for the enthusiasm and encouragement always promptly demonstrated to push the work forward. Also, a big word of appreciation to my co-worker and good friend Sérgio Lopes for his support and help to lighten up the mood in some stressful moments.

Finally, a huge thank to my family and closest friends for the comprehension and comfort given and for always being there when I most needed them.

Acronyms

ACK	Acknowledgment
ABNF	Augmented Backus-Naur Form
ASPAN	Auto configuration and Self-management of Personal Area Networks
BNEP	Bluetooth Network Encapsulation Protocol
BOOTP	Bootstrap Protocol
CLI	Command Line Interface
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
EPI	Election Process Initiator
FQDN	Fully Qualified Domain Name
GN	Group Ad-hoc Network
IANA	Internet Assigned Numbers Authority
IEEE	Institute of Electronic and Electrical Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
L2CAP	Logical Link Control and Adaptation Protocol
LAN	Local Area Network
MAC	Medium Access Control
MAN	Metropolitan Area Network
MRCT	Minimum Route Cost Tree
MST	Minimum Spanning Tree
NAP	Network Access Point
NGN	Next Generation Networks
OFDM	Orthogonal frequency-division multiplexing
OS	Operating System
OSI	Open System Interconnection
PAN	Personal Area Network
PANU	Personal Area Network User
PCP	PAN Control Protocol
PDA	Personal Digital Assistant
PHY	Physical Layer
PoA	Point of Attachment
POS	Personal Operation Space
RFC	Request For Comments
SDP	Service Discovery Protocol
SSID	Service Set Identifier
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System

USB	Universal Serial Bus
UWB	Ultra Wide Band
VoIP	Voice over IP
WAN	Wide Area Network
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

List of Figures

Table of Contents

1 Introduction

In the Next Generation Networks (NGNs) an All-IP communication era will arise. The IP (Internet Protocol) technology will be used as basis to integrate cellular/mobile networks, Wireless Personal Area Networks (WPANs) and Wireless Local Area Network (WLAN) under the same umbrella. Also, IP will be used as the common ground to enable all types of services and applications, such as voice communication (VoIP), video on-demand, e-mail, and web browsing. An evolution towards ubiquitous connectivity will happen and the "Always Best Connected" paradigm will be a must. On the other hand, users will carry Personal Area Networks (PANs) with them instead of multiple stand-alone devices as it happens today. The creation and management of PANs automatically and dynamically and the seamless adaptation of the technology to different networking contexts and user needs will be crucial in the context of NGNs.

Currently, incipient PANs can be created using different technologies. Bluetooth [1] has been used as the standard solution. However, other technologies such as Wi-Fi [2] in ad-hoc mode and the upcoming WiMedia Ultra Wide Band (UWB) [3,[4] also enable the creation of PANs. Still, these are just incipient PANs, in the sense that they can only be set up by means of manual configurations and usually require networking knowledge. In addition, they do not provide intelligent mechanisms that are able to adapt automatically and dynamically to different networking contexts and user preferences/needs. Despite of this, all these WPAN technologies represent enabling technologies for next generation PANs. In addition, all can appear as Ethernet links to the upper layers of the protocol stack, which eases the deployment of IP networks over them. Thus, a new solution that takes all these aspects into account and adds new intelligent adaptive mechanisms that enable the automatic and dynamic adaptation of the PAN to the communication environment and user needs is all that is needed to create a PAN of the future.

Intensive research on this field is being carried out. Ongoing research projects and multiple discussion forums address these topics, and point out solutions; in [5,6,7] some solutions are presented. Campos and Ricardo present a new framework [6,8], the Autoconfiguration and Self-management of Personal Area Networks (ASPAN), which represents a new solution for next generation PANs. ASPAN defines mechanisms for self-creating and self-managing a PAN in the heterogeneous environments envisioned for NGNs, and deals with the dynamic and automatic connection of a PAN to the Internet based on user-defined policies.

The work presented herein represents the implementation of a first prototype of the ASPAN solution. Therefore, it implements a subset of the features provided by ASPAN. Our implementation considers: 1) setup of a common communication base independent from the technology; 2) auto configuration of communication interfaces; 3) a signaling protocol used to create and manage the PAN; 4) election of a master device that is in charge of dynamically manage and configure the PAN; 5) PAN topology discovery during bootstrapping; 6) automatic configuration of a IP (Internet Protocol) network within the

PAN; In this ASPAN prototype the aspects regarding the bootstrapping process, the setup of an IPv4 network over the PAN, the management of topological changes within the PAN the and the dynamics associated with that process were mainly considered.

1.1 Scope

This work fits in the (Wireless) Personal Area Networks domain. A Personal Area Network (PAN) represents an ad-hoc network formed by the interconnection of information technology devices within the range of an individual person, typically within a range of 10 meters. For example, a person travelling with a laptop, a personal digital assistant (PDA) and a portable printer could interconnect them without having to plug anything in, using some form of wireless technology. PANs are formed by wireless communications between devices by means of technologies such as Bluetooth or Ultra Wide Band (UWB). The concept of a PAN was proposed by Thomas Zimmerman and other researchers at M.I.T.'s Media Lab and later supported by IBM's Almaden research lab. In a research paper, Zimmerman explains why the concept might be useful:

"*As electronic devices become smaller, lower in power requirements, and less expensive, we have begun to adorn our bodies with personal information and communication appliances. Such devices include cellular phones, personal digital assistants (PDAs), pocket video games, and pagers. Currently there is no method for these devices to share data. Networking these devices can reduce functional I/O redundancies and allow new conveniences and services.*"

In a near future, PANs are envisioned to have a major role in people's everyday life. The All-IP communication of the NGNs and the services available (VoIP, web browsing, video streaming, e-mail...) will enable the WPANs to grow fast. People will carry a set of electronic personal devices, possibly each one with a different communication technology. To enable the communication between them is a must. Today is already conceivable the idea uploading an mp3 file from an mp3 player to a mobile phone or to stream load a recorded video from a digital camera onto a desktop computer. But, in a few years, we will want to have the possibility to check the status of a pace maker in a PDA or to have the blood's glucose level measured and reported in real-time running a mobile phone application. This enhances the importance or PANs. Currently, incipient PANs can be created using different technologies but usually they involve networking knowledge and manual configuration. A solution that establishes a base for these devices to automatically configure and communicate together is still to be implemented. This work intends to implement a possible solution.

1.2 Project Goals

The main goal of this work was to prototype the ASPAN solution targeting the configuration of a next generation PAN which dynamically adapts to changes in its network environment and takes into account multiple communication technologies. In an attempt to separate the work by phases, the following sub-objectives were defined:

- Study of the technologies and tools to be used in the implementation of the ASPAN solution, such as:
    - IP network and Dynamic Host Configuration Protocol (DHCP) set up
    - IEEE (Institute of Electrical and Electronics Engineers) 802.1D bridges
- Network interface auto configuration.
- Implementation of a signaling protocol over Layer 2 of the OSI (Open Standards Interconnect) model, called PAN Control Protocol (PCP), using Linux packet sockets
- Election of a Master and a Sub Master to manage the topology changes within the PAN due to join/leave of devices or changes with respect to the wireless/wired links forming the PAN
- Implementation of a new spanning tree algorithm for the 802.1D used to configure the active topology of the PAN
- Dynamic management of joining and leaving PAN devices.

1.3 Major Results

This section identifies the major results achieved along the work. A separate section is devoted to each result for the sake of clearness.

1.3.1 ASPAN Prototype dealing with Intra-PAN Connectivity

The major result is the deployment of an ASPAN prototype, dealing with intra-PAN connectivity issues, based on the ASPAN solution developed by Campos and Ricardo. However, in order to have this major result some partial results were obtained along the work. Those are listed below.

1.3.1.1 Communication Interface Auto-configuration System

This system allowed relegating the user from any manual configurations. The simultaneous presence of multiple communications technologies and their individual characteristics turn this into a non-trivial process. The connection configurations (Bluetooth, Wi-Fi and Ethernet) and the bridge setup are done automatically. This enables the prototype to run as a background management service in the user's devices.

1.3.1.2 PAN Control Protocol Implementation

The PCP is a protocol based in the exchange of Layer 2 messages between the members of the PAN, allowing communication between devices regardless of the existence of an IP network and independently of the underlying wireless/wired technologies (Wi-Fi, Ethernet, Bluetooth…).

1.3.1.3 Master and Sub Master Election Mechanism Implementation

ASPAN is based on a master-slave model. When devices come together to form a PAN one of them is elected as the master using the election algorithm; this is the very first step performed at the connection level. Afterwards, the Master is in charge of managing the connectivity aspects of the PAN. Every time a new device joins or leaves the PAN such

information is reported to the Master so that it can take the proper decisions regarding intra-PAN topology. The Sub Master acts as fail-safe device. Having the same information about the PAN as the Master, if the latter disconnects or leaves the PAN the Sub Master takes its place without notice from the other PAN devices' perspective.

1.3.1.4 Spanning Tree Algorithm implementation

The spanning tree algorithm, *Campos's algorithm*, [9] is at the core of the PAN managing process. During the join procedure the Master gathers information regarding how is the new peer connected to the PAN, i.e. which technologies are used and to each PAN members it is connected to. Using this information the PAN Master can decide about the best PAN topology setup and then apply it by instructing all other peers to act accordingly.

1.3.1.5 Creation of an IPv4 Network

The creation of an IPv4 is the final stage of the PAN configuration. This is done using an auto configuration mechanism, namely DHCP. The Master runs the DHCP Server and all other peers run the DHCP Client to acquire a valid IP address and corresponding network settings. Afterwards, the PAN devices can communicate with each other using any IP application to share services such as audio streaming or file transfer.

1.3.1.6 Management of Arrival of New PAN Devices

The management of arrival of new PAN devices includes two mechanisms: the join mechanism and the guest mechanism.

In the join mechanism, the new PAN device announces its arrival to the network and the PAN Master collects the information about it including its location in the PAN. The Master then adjusts the active topology according to the output of the spanning tree algorithm and informs the remaining peers about their new bridge setup. Also, the arriving peer will be asked by the Master to run the pre-determined auto-configuration mechanism (DHCP) to obtain a valid IP network address. The new peer becomes able to communicate with all other PAN devices using the IP subnet defined for intra-PAN connections and, if available by any peer, connect to the Internet.

The guest mechanism, implements the support for legacy devices. In this case, the arriving device is not running the ASPAN software and may be using other operative system. However, the user is in charge of handling all the connection setup which may be more or less complex depending on the technology used to create such connection. Afterwards, the user runs the DCHP client on the device to obtain an IP address and take advantages of the PAN features, such as Internet access (if available). It is important to realize that from the PAN peers point of view the legacy device is totally invisible. Moreover, the PAN Master is not aware of the legacy device and, as such, does not consider it to define the active PAN topology.

1.3.1.7 Managing the Departure of Devices from the Network

This procedure has two separate functions. The first involves notifying the neighbours of the leaving device so that they are able to reconfigure their interfaces and set them up to

listen for new PAN devices. The second one consists in notifying the PAN Master. This way the PAN Master can re-compute the proper active topology (now excluding the leaving device) and notify the other PAN devices about the new setup. If the leaving device is the Master then it announces its exit to the Sub-Master which auto-elects itself, becomes the PAN Master and elects a new Sub Master. On the other hand, if the leaving device is the PAN Sub-Master only a new PAN Sub-Master is elected.

1.4 Structure of the Dissertation

This report is divided into seven chapters. Chapter 2 presents the state of the art on wireless technologies, in particular wireless PAN technologies and the existing solutions dealing with the automatic and dynamic interconnection of devices within ad-hoc networks. Chapter 3 describes the ASPAN solution used as basis for our implementation and provides the theoretical background for our work. Chapter 4 mentions the software tools used in the implementation as a complement to the developed software. Chapter 5 presents the work carried out and the obtained results. Chapter 6 evaluates the developed ASPAN prototype. Chapter 7 draws the conclusions and refers to future work.

2 State of the Art

There are some important technologies that should be referred so that we can understand the importance of this project in the times that run by. A Wide Area Network (WAN) is a computer network that covers a broad area. It is a network whose communication links cross metropolitan or national boundaries. The largest and most well-known example of a WAN is the Internet. Metropolitan Area Networks (MAN) are smaller than WANs but large enough to span a city area. Wireless MAN defines broadband Internet access fixed or mobile devices via antennas in a city. A Local Area Network (LAN) is a computer network covering a small geographic area, like a home, office or group of building. It is often used as a private network. Current Wireless LANs are most likely to be based on IEEE 802.11 technology [2]. PANs are the smallest computer networks and its reach is typically a few meters. They are formed by devices close to one person.

There are multiple wireless technologies that can be useful to PANs like UWB, Wi-Fi and Bluetooth. In the following sections some of the prominent wireless technologies are presented, with special focus on wireless PAN technologies used as enabling technologies in our work. Also, some routing ad-hoc protocols that present alternate solutions to interconnect personal devices will be presented.

2.1 Bluetooth

Bluetooth is an industrial specification for wireless personal area networks. Bluetooth provides a way to connect and exchange information between devices such as mobile phones, laptops, PCs, printers, digital cameras, and video game consoles over a globally unlicensed short-range radio frequency. This technology is acceptable for situations when two or more devices are in proximity to each other and don't require high bandwidth. It also simplifies the discovery and setup of services. Bluetooth devices advertise all services they provide. This makes the utility of the service much more accessible, without the need to worry about network addresses, permissions and all the other considerations that go with typical networks. Bluetooth allows simultaneous bidirectional communication ("full duplex") or sequential ("half duplex"). In the former case data rates can go up to 400 kilobits per second in both directions while in the latter transmition can go up to 721 kilobits per second in one direction and 57,6 kilobits per second in the other.

2.1.1 Bluetooth PAN Profile

Using Bluetooth wireless technology, devices have the ability to form networks and exchange information. For these devices to interoperate and exchange information, a common packet format has been defined to encapsulate Layer 3 network protocols. That protocol is the Bluetooth Profile version 1 [10] and is used to encapsulate IP packets over BNEP (Bluetooth Network Encapsulation Protocol) headers. BNEP is used to transport common networking protocols over the Bluetooth media such as IPv4 and IPv6. The packet format is based on EthernetII Framing as defined by IEEE 802. BNEP runs over L2CAP

and reuses the Ethernet packet format commonly used for local area networking technology. For this profile, three general scenarios are discussed: Network access points, Group Ad-hoc Networks, PANU-PANU (PAN user). Each of the scenarios has unique network architecture and unique network requirements, but all are various combinations of a PAN. A network access point is a unit that contains one or more Bluetooth radio devices and acts as a bridge, proxy, or router between a Bluetooth network and some other network technology. Group ad-hoc networking allows mobile hosts to cooperatively create ad-hoc wireless networks without the use of additional networking hardware or infrastructure. A point to point connection between two PANUs allows direct communication between these two nodes only.

2.2 Wi-Fi (IEEE 802.11)

Wi-Fi is a wireless technology standard defined by the IEEE 802.11-1997 standard and then clarified in IEEE 802.11-1999 standard. Nevertheless, it was only its successor 802.11b, which was accepted as the definitive wireless LAN technology. 802.11b uses the 2.4GHz frequency spectrum with a bandwidth of 11Mbps versus the 2Mbps provided by the original standard. There are other 802.11 technologies, including 802.11a (5GHz, 54Mbps) and 802.11g (2.4Ghz, 54Mbps). Wi-Fi continues to be the pre-eminent technology for building general-purpose wireless networks. Wi-Fi has become as prevalent a technology at home as it is at work. Clearly, Wi-Fi will continue to be an important force in the market for some time to come. Even though it was designed primarily for private applications, WiFi is also being deployed in public places to create so-called hotspots, where WiFi-capable users can obtain broadband Internet access. This new domain of application could be the major future market opportunity for WiFi, but in order to take advantage of it, several key challenges, both technical and business-related, must be overcome.

Wi-Fi differs from Bluetooth in that the former provides higher throughput and covers greater distances, but requires more expensive hardware and higher power consumption. They use the same frequency range, but employ different multiplexing schemes. While Bluetooth is a cable replacement for a variety of applications, Wi-Fi is a cable replacement only for local area network access. Bluetooth is often thought of as wireless USB, whereas Wi-Fi is wireless Ethernet, both operating at much lower bandwidth than the cable systems they are trying to replace. However, this analogy is not entirely accurate since any Bluetooth device can, in theory, host any other Bluetooth device - something that is not universal to USB devices.

2.3 WPAN (IEEE 802.15)

IEEE 802.15 is the 15th working group of the IEEE 802 which specializes in Wireless PAN Standards. The IEEE 802.15 standard defines physical layer (PHY) and medium access control (MAC) specifications for wireless connectivity with fixed, portable, and moving devices within or entering a personal operating space (POS). A POS is the space about a person or object that typically extends up to 10 m in all directions and envelops the

person whether stationary or in motion. The original goal of the IEEE 802.15.1 Task Group was to achieve a level of interoperability that could allow the transfer of data between a WPAN device and an IEEE 802.11 device. Although this proved infeasible, IEEE Std 802.15.1-2005 does have mechanisms defined to allow better coexistence with IEEE802.11b class of devices. WPAN standards include:

- 802.15.1a: Standardizes Blueetoth's MAC and PHY levels (2.4GHz at 1Mbps)
- 802.15.2: Coexistence of PANs with one another
- 802.15.3: High rate PAN, used for UWB (2.4GHz at 55 Mbps)
- 802.15.3a: Alternative high rate PAN for UWB (2.4GHz at 110 Mbps)
- 802.15.4: Low rate PAN - Standardizes ZigBee's MAC and PHY levels
- 802.15.4a: Alternative low rate - low power UWB

2.4 Ultra Wide Band

Ultra-Wideband (UWB) [3,[4] is a technology for transmitting information spread over a large bandwidth (>500 MHz) that should, in theory and under the right circumstances, be able to share spectrum with other users. This is intended to provide an efficient use of scarce radio bandwidth while enabling both high data rate personal-area network (PAN) wireless connectivity and longer-range, low data rate applications as well as radar and imaging systems. Due to the extremely low emission levels currently allowed by regulatory agencies, UWB systems tend to be short-range and indoors. However, due to the short duration of the UWB pulses, it is easier to engineer extremely high data rates, and data rate can be readily traded for range by simply aggregating pulse energy per data bit using either simple integration or by coding techniques. Conventional orthogonal frequency-division multiplexing (OFDM) technology can also be used subject to the minimum bandwidth requirement of the regulations. High data rate UWB can enable wireless monitors, the efficient transfer of data from digital camcorders, wireless printing of digital pictures from a camera without the need for an intervening personal computer, and the transfer of files among cell phone handsets and other handheld devices like personal digital audio and video players.

As a way to allow Ethernet packets to be sent over UWB, the WiMedia Alliance is in the process of publishing the WiMedia Networking Protocol currently known as WiNet. WiNet defines a Logical Link Control Layer networking protocol for the WiMedia radio platform to model the behavior of an IEEE 802 environment. Since IEEE 802 is the basis of for both Wi-Fi (IEEE 802.11) and Ethernet (IEEE 802.3), WiNet is designed to support easy bridging between these networks. The draft standard proposes IEEE 802.1D bridges to integrate UWB networks with the IEEE 802-like networks.

In the next figure we can see where WiNet will stand compared to the already established technologies.

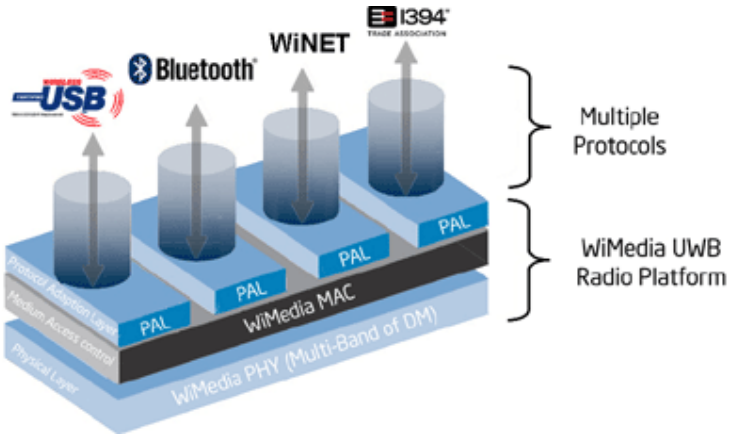

Figure 1 – UWB architecture integrating multiple technologies [20]

It is believed that UWB, with its technical and economic advantages, should help enable mainstream adoption of WPANs. Some people envisions a world of pervasive wirelessly connectivity in the home and in the office for all. UWB can help to achieve such vision.

2.5 ZigBee (IEEE 802.15.4)

ZigBee is the name of a specification for a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). ZigBee is targeted at RF applications that require a low data rate, long battery life, and secure networking. Their protocols are intended for use in embedded applications requiring low data rates and low power consumption. ZigBee's current focus is to define a general-purpose, inexpensive, self-organizing, mesh network that can be used for industrial control, embedded sensing, medical data collection, smoke and intruder warning, building automation, home automation, domotics, etc. The resulting network will use very small amounts of power so individual devices might run for a year or two using the originally installed battery. This technology is not designed for the PAN scenario of this work. It is more adequate to use it in sensor networks.

2.6 Ad-hoc On-demand Distance Vector

The Ad hoc On Demand Distance Vector (AODV) algorithm [21] is a routing protocol designed for Mobile Ad-hoc Networks (MANETs) and other wireless ad-hoc networks. AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

2.7 Optimized Link State Routing Protocol

The Optimized Link State Routing Protocol (OLSR) [22] is developed for Mobile Ad-hoc Networks. It operates as a table driven and proactive protocol, thus exchanges topology information with other nodes of the network regularly. The nodes which are selected as a multipoint relay (MPR) by some neighbour nodes announce this information periodically in their control messages. Thereby, a node announces to the network, that it has reachability to the nodes which have selected it as MPR. In route calculation, the MPRs are used to form the route from a given node to any destination in the network. The protocol uses the MPRs to facilitate efficient flooding of control messages in the network. OLSR inherits the concept of forwarding and relaying from HIPERLAN (a MAC layer protocol) which is standardized by ETSI.

In conclusion, it is easy to understand that there is still a lot of work to be done in this area! Despite that, things are improving fast and something like this project could speed up things a little by trying to "join" some of these technologies that, until now, were apart. As it was already said before, this work was made using the Bluetooth and WLAN (Wi-Fi in a LAN) technologies, mainly because these are the most important in the market at the time. The referred MANET protocols present solutions at Layer 3 (OSI Model) which imply the definition of new IP auto configuration mechanisms and prevent the transparent use of announcement protocols such as Universal Plug and Play (UPnP) which assume the existence of only one logic link. Having a unique logic link enables the reutilization of other protocols defined for LANs in PANs and allows the use of multiple Level 3 protocols (IPv4, IPv6). Therefore the presented MANET protocols are not suited for PANs.

3 Theoretical Background

There is not much background theory besides the ASPAN framework and the background theory of PANs. In this chapter the theoretical background needed to implement the ASPAN solution will be described. The main background needed in this work is the ASPAN Framework in which the fundaments of this solution are stated as well as the packet format used to implement the signalling protocol in the development of the ASPAN solution. Finally a description of the DHCP protocol, one of the standard protocols nowadays used for automatic creation of IP networks, will be presented followed by the IEEE 802.1D protocol for bridging network interfaces widely used in the implementation . A NGN example is shown in Figure 2. Multiple technologies and devices are present including Bluetooth and WLAN used in our work.

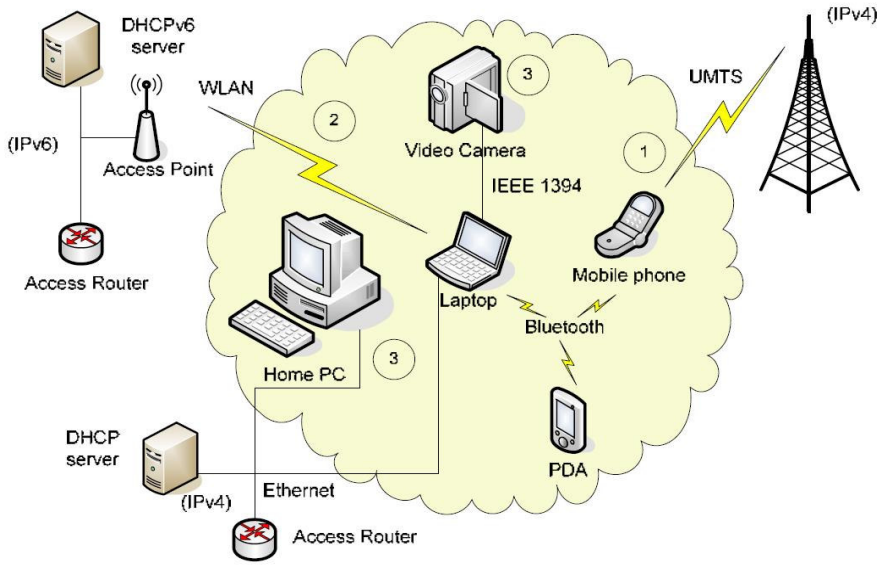

Figure 2 – Example scenario for Next Generation Networks from end-user perspective [23]

3.1 The ASPAN Framework

The Auto-configuration and Self-management of Personal Area Networks (ASPAN) [6,[8] is the base framework used in this work. ASPAN has been proposed by Campos and Ricardo and aims at addressing the auto-configuration and self-management of a Personal Area Network (PAN) in a NGN communication scenario. In the following sections, the main features of the framework are presented.

3.1.1 Master-Slave Paradigm

ASPAN is based on a master-slave model. When devices come together to form a PAN one of them is elected as the master of the network using the election algorithm presented in Section 3.1.2.1. The master device is in charge of collecting information from all slaves and it stores locally topological information of the PAN. It also deals with new

devices connecting to the PAN, as well with devices leaving the network. Furthermore, it enables the ASPAN to inform the user about the set of terminals currently participating in the PAN and the way they are organized; dissemination of this information to the other devices of the PAN can be performed on-demand, i.e., every time a slave requests it explicitly.

When the current master leaves the PAN, a new master needs to be elected. In order to deal with this problem, a sub-master device is also elected during the master election process. The sub-master mirrors all context information required to maintain the PAN, such as the topology of the PAN. The sub-master is always synchronized with the master; that is, every time the master detects a modification in the context, for instance, some device left the PAN, it notifies the sub-master. The sub-master operates in background as a fallback device (i.e., has a passive role) so that when the master (suddenly) leaves the PAN the sub-master can assume the master role immediately, and the PAN does not get master orphan. Three scenarios are possible:

1. Sub-master becomes master by nomination – current master nominates the sub-master as the new master of the PAN. The selection of a new sub-master between the remaining PAN devices needs to be performed by using the same criteria used to elect the current master. However, this election is performed by the new elected master based on the information it stores locally about each device belonging to the PAN, which reflected in the current topological tree. In addition, the new master announces itself within the PAN as the new master.

2. Sub-master auto-elects itself as new master – current sub-master detects that the master is no more connected to the PAN, and auto-elects itself as the new master of the PAN. Afterwards, it elects a new sub-master and announces itself within the PAN as the new master.

3. Master and sub-master need to be re-elected from the scratch – both master and sub-master leave the PAN without notice. In this situation, the two previous alternatives cannot be applied. Then, a new master and sub-master need to be elected between the devices still connected to the PAN using the same approach that was used when the PAN was created at the very beginning, that is, the PAN needs to be created from the scratch.

3.1.2 Master Election and Topology Discovery

This section describes the mechanism and the algorithm used for electing the master and for discovering the topology of the PAN. The master is only elected when there are at least two devices forming the PAN. In the following, first the election algorithm and the parameters taken into account to elect the master are described. Afterwards, the election and topology discovery mechanism is defined.

3.1.2.1 Master Election Algorithm

The election of the master is performed based on an algorithm that considers four parameters:

- Battery capacity (mWh)
- CPU capability (MHz)

• Memory capacity (MB)
• Number of network interfaces

ASPAN considers weights associated to each of these parameters in order to account for parameters that may have higher importance than others. For that purpose, ASPAN defines the following coefficients associated to each parameter:

• W_{bat} – weight associated to the battery capacity parameter
• W_{CPU} – weight associated to the CPU capability parameter
• W_{mem} – weight associated to the memory capacity parameter
• W_{netif} – weight associated to the number of network interfaces parameter

The values assigned to each of these coefficients depend on the relevance assigned to each parameter. If all coefficients have the same relative relevance, then:

$W_{bat} = W_{CPU} = W_{mem} = W_{netif} = 0.25$

From now on, we consider that all parameters have the same relevance.

The election algorithm works as follows. As soon as the device that has started the election process collected all required information from its partners, using the mechanism explained in the next section, it will run the election algorithm. For election purposes each PAN device is modeled as a cartesian point in the 4-dimensional Cartesian space defined by the parameters presented above. Thus, for instance, a Laptop can be modeled by the following cartesian point:

Laptop = (battery capacity, CPU capability, memory capacity, nº of network interfaces)
= (W_{bat}, W_{CPU}, W_{mem}, W_{netif})

Given the points representing each PAN device, the master is the device which distance to the origin of the 4-dimensional cartesian space is the highest. From now on, this distance will be referred as BIMP (which stands for Battery, Interfaces, Memory and Processor). Each PAN device computes its BIMP and reports it back to the device that initiated the election process which, in turn, will inform the elected device using the mechanism described in the next section.

3.1.2.2 ASPAN Master Election and Topology Discovery Mechanism

At the very beginning, when PAN devices come together to create the PAN the topology of the network is unknown. The election process requires that all nodes get visited in order to collect information needed to run the election algorithm explained in previous section. Therefore, the mechanism used for collecting such information can be used to collect information about the topology of the network; this is exactly what we consider herein.

1. The device initiating the process (initiator) broadcasts an Election message towards its neighbours, i.e., the nodes connected to the links it is connected to
2. A node receiving an ELECTION message elects the sender of that message as its parent node from the election mechanism point of view. The parent device

is the device to which the current device has to return the topological information it could found out while interacting with its own neighbours.

3. Upon receiving an ELECTION message, and if not yet participating in any election process, the current device re-broadcasts the message to its own neighbours.
4. After receiving the election message the device replies with an ACK message to the election starter. This message already includes its distance to the origin of the 4-dimensional cartesian point.
5. The election mechanism finishes when the initiator has received ACK messages from all its neighbours.

It is worth noting that, while collecting information from the PAN devices used for running the election algorithm at the initiator, this mechanism is able to collect the topological information from the network. Then, at this point, the initiator has information about all devices currently connected to the PAN and the way they are connected to each other (topological information). This information includes the MAC addresses of each node interface, the MAC address it's connected to and the type of technology used in this connection. At this point, the topological information consists of a set of branches that combined form the connectivity graph modelling the topology of the PAN.

The different "election" and "ack" messages and their route in an election process are displayed in Figure 3. In this example we assume six devices within the PAN being the device number two the one to start the election process.

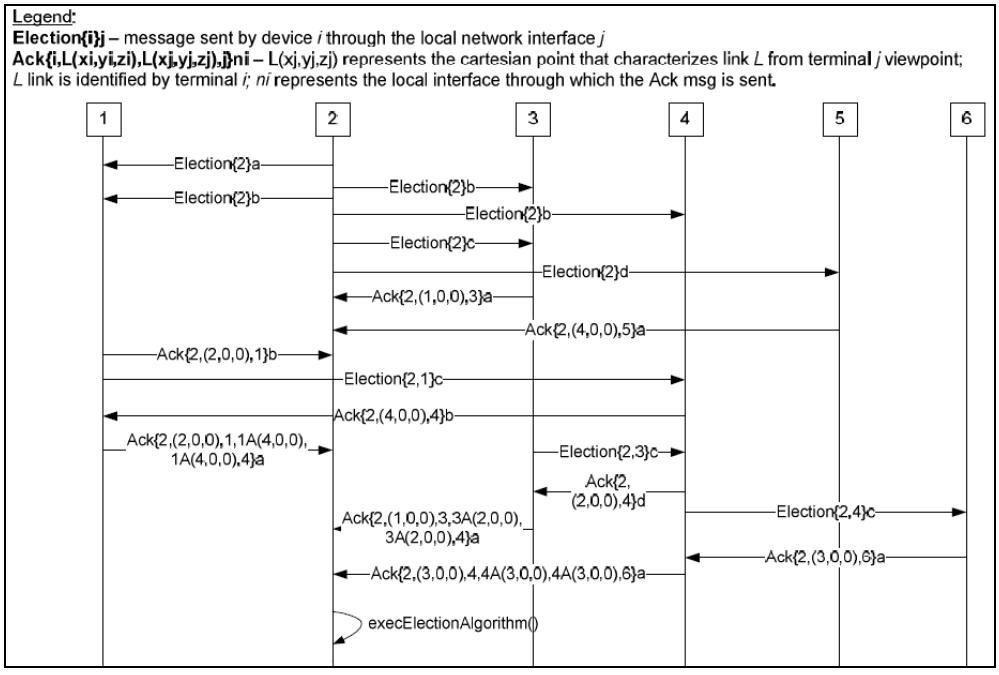

Figure 3 – Illustration of an election process

3.1.3 Device Identification within the PAN

In the ASPAN scope the identification of PAN devices is performed by reusing the identification strings already defined by legacy technologies, such as Bluetooth and WLAN. The identification of a device is always referred to the local domain defined by the PAN, whose composition is specified by the user, which identifies the terminals belonging to his PAN; for instance, JohnsPAN and BobsPAN identify possible IDs for the PAN local domain. For that purpose, the Service Set Identifier (SSID) and the Bluetooth Device Name parameters are reused. In fact, either WLAN or Bluetooth do not specify any concrete syntax for these identification strings. Thereby, we are free to specify the most suitable syntax for our applicability domain; we use the syntax associated to a Fully Qualified Domain Name (FQDN) used in the Internet for device identification. The same identification is used regardless the specific Link Layer technology being used. The identification string (or FQDN) is mapped into both a MAC address and IP address (when IP connectivity is already established).

3.1.4 Joining Procedure

When a user's device is joining the PAN, it firstly discovers the device(s) of the PAN in the neighbourhood by using the proper Layer 2 mechanisms. This new device can only find the PAN if there's some device in it using the same kind of technology. After detecting the presence of one or more devices, the joining device broadcasts a message requesting to join the PAN. The device which receives the election message knows that the PAN is already created and forwards the election message to the PAN Master. The PAN Master has now several ways to act. Firstly, it checks which will be the rank of the new device. There are three possibilities: Master, Sub-Master, or Peer. According to this, the proper message is sent back to the new device. Also, and simultaneously, the PAN Master adds the new link information provided by the arriving device and re-computes the proper active topology for the PAN. Then it communicates the new setup information to all PAN devices.

3.1.5 Leaving Procedure

Every device connected to the PAN regularly scans its link. When it detects that some partner in that link has left the PAN, it notifies the master which knows the PAN topology. Using that information, the master updates the active topology accordingly. Also, it notifies its slaves about the current event so that each slave can update its local configurations, for instance, the links that it should activate to make part of the active topology. If the leaving node is the master device, the sub-master is elected as the new master and a new sub-master is elected. On the other hand, if it is the sub-master that is leaving the PAN, a new sub-master is just elected and the PAN can still operate without any change. Finally, if both the master and sub-master leave the PAN at the same time, the PAN has to be created from the scratch by the remaining PAN devices.

3.1.6 Network Configuration

Network configuration in the context of the ASPAN framework is divided in two components:

1. Configuration of specific PAN devices as interconnection nodes between different links of the PAN;
2. Configuration of IP addresses and optional information for each PAN device for enabling IP connectivity within and to the outside of the PAN.

These two components are explained in detail in the two following sections.

3.1.6.1 Configuration of Devices acting as Interconnection Nodes

In multi-hop scenarios the configuration of devices for enabling connectivity between the multiple devices composing a PAN is performed using bridging, specifically IEEE 802.1D bridges. Using bridging to interconnect devices connected to different links within the PAN results in the multiple devices becoming connected to the same logical link. This eases the deployment of traditional auto-configuration mechanisms, such as DHCP. Furthermore, it enables the creation of a single IP sub network to which all PAN devices can be connected; this also eases the operation of traditional mechanisms, such as Address Resolution Protocol (ARP) [11] and Neighbour Discovery Protocol (NDP) [12]. On the other hand, even from the ASPAN framework, this type of PAN configuration can ease the transmission of control messages between PAN devices, since there is direct connection from every device to every device; concerning notification of some event the PAN Master just has to broadcast it to the local link and all slaves get informed.

3.1.6.1.1 Technology-dependent Aspects

3.1.6.1.1.1 Bluetooth

When deploying the PAN using Bluetooth technology the Bluetooth PAN profile is considered. This profile defines three scenarios and three related roles for the Bluetooth devices: Gateway Node (GN), Network Access Point (NAP), and PAN User (PANU). When a new Bluetooth device using "mobile.BobsPAN" as ID string tries to join/create a PAN it will scan looking for any device with an ID string, whose suffix is "BobsPAN". If such device is found an election/join message is sent to the existing device(s), otherwise this device will remain in a "wait election" state.

3.1.6.1.1.2 Wireless LAN

Similarly to Bluetooth, the WLAN technology specifies the use of an identification string (SSID). Nevertheless, rather than in Bluetooth, where this string is used for informational purposes only, in WLANs it is used for distinguishing between different WLAN networks. Therefore, the ASPAN solution suggests the following configurations:

- All PAN devices that are not connected to the PAN are configured with their own identification strings which, in this case, are mapped to the WLAN SSID; such as in Bluetooth, the FQDN syntax is employed, e.g., "Laptop.BobsPAN."
- After finding out some neighbour PAN device, the current PAN device changes the SSID of its local WLAN network interface from its own identification string to the identification string of the PAN, for instance, from "Laptop.BobsPAN."

to "BobsPAN."; every PAN device does this in order to form a single WLAN network with the neighbour PAN devices.

In terms of software implementation some changes were made in order to overcome the slow reconfiguration time of a WLAN network card. This was done in the second stage of the work, as described in section 5.2. Instead of changing the SSID from "Laptop.BobsPAN" to "BobsPAN" the SSID is always equal to the identification string selected for the PAN, in this case "BobsPAN". Due to slow reconfiguration time, changing the SSID would cut the connection between both devices and cause the loss of control messages. Also, with both devices trying to create a network with the same name at the time it may happen that they create two separate networks with different AP/Cell addresses..

3.1.7 Configuration of IP Connectivity within a PAN

In the beginning, when the PAN devices are forming a PAN, IP connectivity does not exist yet. Upon forming the PAN or joining it, PAN devices configure an IP address for intra-PAN communication. This local address is intended to be used within the PAN and its configuration depends on PAN logical topology. Since the PAN devices are assumed to be all connected to the same logical link, the traditional auto-configuration mechanisms can be used for intra-PAN IP address configuration.

3.1.8 ASPAN's Spanning Tree Algorithm

The use of IEEE 802.1D bridges regarding the interconnection of multiple links within the PAN requires the definition of a single spanning tree as the active topology of the PAN. In this section we describe the spanning tree algorithm defined by the ASPAN framework to select such spanning tree. Two major advantages come up: 1) a single logical link is created within the PAN; 2) global optimization is achieved instead of optimization for each participating device like it happens when using routing protocols. The major disadvantage has to do with the rejection of some links that could potentially be used for intra-PAN connectivity. Nevertheless, in the context of future PANs the number of suitable links that are rejected will be minimal. On the one hand, because the number of redundant links within a PAN is not envisioned to be high; on the other hand, because the coexistence of heterogeneous links with the same PAN makes some of them undesirable to make part of the active topology. In addition, the only case where this disadvantage comes up is when the user is willing to optimize intra-PAN connectivity; this is not always the case. Thus, upon electing the master and the sub-master the links that will belong to the intra-PAN spanning tree and the devices that will be configured as bridges between different link technologies are selected. Although the ASPAN framework predicts several user-defined policies such as optimization for external connectivity, optimization for internal connectivity, optimal power consumption or any possible combination between the three, in this prototype phase the internal connectivity has been chosen as default user policy therefore each link is defined by its data rate.

The algorithm used to calculate the best PAN topology is the Campos's algorithm [9] which represents a new approximate Minimum Routing Cost Tree (MRCT). This algorithm

improves the performance of bridged Layer 2 networks, particularly when the edges of the graph modelling the network have heterogeneous weights. It aims at improving the routing cost of the active spanning tree assumed in these networks, taking into account that the MRCT is by definition the optimal spanning tree. The resulting spanning tree has lower routing cost than the spanning tree computed using the spanning tree algorithm currently used by IEEE 802.1D bridges.

3.2 The Messages Packet Format

To communicate between the PAN devices a standard message packet was previously defined. The entire PCP messages are exchanged in the data link layer using the Ethernet protocol and the Ethernet v2 framing REF (see Figure 4).

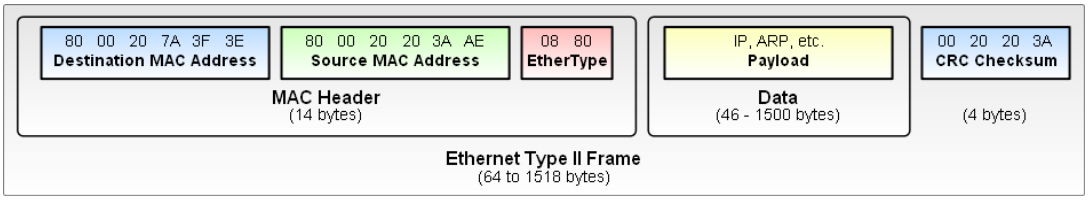

Figure 4 - Ethernet type 2 framing

Using Linux packet sockets with "SOCK_RAW" as socket type it is possible to create raw packets including the link level header. This allowed us to fully customize the PCP messages. In the EtherType field of the MAC header the number 0x0880 was chosen to represent the PCP. Any number above 0x05DC (1.500 decimal) was suitable (numbers below remote to original Ethernet framing format where this field indicated the packet lenght). Also, according to the IEEE EtherType Registration Authority [13], 0x0880 is not currently reserved by any company or organisation. The CRC Checksum was implemented to enable the detection of errors in incoming packets. A CRC is an error-detecting code whose computation resembles a long division computation in which the quotient is discarded and the remainder becomes the result, with the important distinction that the arithmetic used is the carry-less arithmetic of a finite field. The length of the remainder is always less than the length of the divisor, which therefore determines how long the result can be. In our messages the CRC-32-IEEE 802.3 was used because it is the standard CRC used in Ethernet messages.

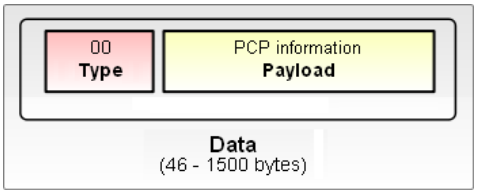

Figure 5 - PCP message format

About 20 different messages are used by PCP. To easily determine which kind of message is arriving, the first byte of the Data field is used to distinguish between them as shown in Figure 5 above. A full-detailed list of the PCP messages is present in Annex A.

3.3 The DHCP Protocol

Dynamic Host Configuration Protocol (DHCP) [14] is an auto configuration protocol used by networked computers (*clients*) to obtain IP addresses and other parameters such as the default gateway, subnet mask, and IP addresses of DNS servers from a DHCP server. The DHCP server ensures that all IP addresses are unique, e.g., no IP address is assigned to a second client while the first client's assignment is valid (its *lease* has not expired). Thus IP address pool management is done by the server and not by a human network administrator. DHCP uses the same two Internet Assigned Numbers Authority (IANA) assigned ports as BOOTP [15]: 67/udp for the server side, and 68/udp for the client side. DHCP operations fall into four basic phases. These phases are IP lease request, IP lease offer, IP lease selection, and IP lease acknowledgement. After the client obtained an IP address, the client may start an address resolution query to prevent IP conflicts caused by address poll overlapping of DHCP servers.

DHCP discovery

The client broadcasts on the local physical subnet to find available servers. Network administrators can configure a local router to forward DHCP packets to a DHCP server on a different subnet. This client-implementation creates a UDP packet with the broadcast destination of 255.255.255.255 or subnet broadcast address.

DHCP offers

When a DHCP server receives an IP lease request from a client, it extends an IP lease offer. This is done by reserving an IP address for the client and sending a DHCPOFFER message across the network to the client. This message contains the client's MAC address, followed by the IP address that the server is offering, the subnet mask, the lease duration, and the IP address of the DHCP server making the offer.

DHCP requests

When the client PC receives an IP lease offer, it must tell all the other DHCP servers that it has accepted an offer. To do this, the client broadcasts a DHCPREQUEST message containing the IP address of the server that made the offer. When the other DHCP servers receive this message, they withdraw any offers that they might have made to the client. They then return the address that they had reserved for the client back to the pool of valid addresses that they can offer to another computer. Any number of DHCP servers can respond to an IP lease request, but the client can only accept one offer per network interface card.

DHCP acknowledgement

When the DHCP server receives the DHCPREQUEST message from the client, it initiates the final phase of the configuration process. This acknowledgement phase involves sending a DHCPACK packet to the client. This packet includes the lease duration and any other configuration information that the client might have requested.

In the PAN, running the DCHP protocol is last step of its setup after which the IP configuration will be done allowing peers to communicate over the network layer.

3.4 Bridges IEEE 802.1D

Bridging is a software forwarding technique used in computer networks. IEEE 802.1D bridges [16] make no assumptions about where in a network a particular address is located. Instead, they use broadcasting to locate unknown devices. They learn the workstation locations by analyzing the source address of incoming frames from all attached networks. Once a workstation has been located, its location is recorded in a local table where the MAC address is stored with its IP Address to avoid unnecessary broadcasts. This is known as the MAC learning process. Due to the need to broadcast, bridging utility is limited to relatively small scale networks, such as LANs or PANs; otherwise network flooding would be a problem. Transparent bridging is the most used in Ethernet networks. It refers to a form of bridging "transparent" (at Layer 3) to the devices using it in the sense that the devices operate as if the bridge is not there. Bridges broadcast between networks, and only allow specific addresses to pass through the bridge to the other network. Also, they can connect networks that deal with different packet formats.

The spanning tree protocol (STP) provides a loop free topology for any bridged LAN. This is essential for bridging, so that the packets do not get lost forever in the loops. The STP is defined in the IEEE Standard 802.1D. STP finds a spanning tree within a network formed by the installer, whether purposefully created or not, and disables the links not part of that tree.

In this chapter the ASPAN framework, the message packet format, the protocol used for setting up an IPv4 network (DHCP), and IEEE 802.1D bridges were presented. This concludes the necessary background theory for our work.

4 Software Tools Used Along the Work:

This chapter refers to some of the software programs used to create the prototype of the ASPAN solution. The prototype was implemented in Linux OS (Ubuntu distribution). To obtain some key properties of the system some external programs were used. Most of these programs are open-source and the needed features they offer could be implemented in the solution but at this stage of the project it was decided not to do that. Instead, these programs remain external to the solution and are used by it when necessary. Section 4.1 describes the external programs used by ASPAN prototype and Section 4.2 mentions the software tools used to test it.

4.1 Standard UNIX Built-in Programs

This section lists the UNIX programs used by the ASPAN solution and also a screenshot of these programs showing their respective output is presented.

4.1.1 *ifconfig*

It was used to configure the kernel-resident network interfaces. Using it is possible to retrieve the status of a desire network interface and/or configure it.

```
root@joao-desktop:/home/joao# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:6E:89:BD:C2
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:185

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:272 (272.0 b)  TX bytes:272 (272.0 b)

nas0      Link encap:Ethernet  HWaddr 00:0E:50:6D:3A:69
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:50ff:fe6d:3a69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:102 errors:0 dropped:2 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8085 (7.8 KiB)  TX bytes:1434 (1.4 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:87.196.24.208  P-t-P:212.0.167.185  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1492  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:518 (518.0 b)  TX bytes:99 (99.0 b)

root@joao-desktop:/home/joao# 
```

Figure 6 – *ifconfig* output

4.1.2 *iwconfig*

It is similar to *ifconfig*, but is dedicated to wireless networking interfaces. It was used to set the parameters of the network interface which are specific to the wireless operation (eg.

SSID, mode, frequency…). *iwconfig* may also be used to display those parameters, and the wireless statistics (extracted from /proc/net/wireless).

```
root@joao-desktop:/home/joao# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

ra0       RT2400PCI  ESSID:off/any
          Mode:Managed  Channel=1  Bit Rate:11 Mb/s
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  invalid crypt:0  invalid misc:0

sit0      no wireless extensions.

nas0      no wireless extensions.

ppp0      no wireless extensions.

panbr     no wireless extensions.

root@joao-desktop:/home/joao#
```

Figure 7 - *iwconfig* output

4.1.3 *hciconfig*

It is used to configure Bluetooth devices. *hciX* is the name of a Bluetooth device installed in the system. Information such as interface type, BD address, ACL MTU, SCO MTU, flags (up, init, running, raw, page scan enabled, inquiry scan enabled, inquiry, authentication enabled, encryption enabled) are also retrieved by *hciconfig*.

4.1.4 *hcitool*

It is used to configure Bluetooth connections and send some special command to Bluetooth devices. Its two most important commands for the ASPAN solution are: 1) "cc" which creates a baseband to a remote device with a given Bluetooth address. 2) "scan" which inquires for remote devices and prints their information.

4.1.5 *iwlist*

It is used to display some additional information from a wireless network interface that is not displayed by *iwconfig*. The *iwlist* is primarily used to generate a list of nearby wireless access points and their MAC addresses and SSIDs. Using this tool it is possible to determine if the PAN is created or if this network interface is the first one in the (still not created) PAN.

4.1.6 *brctl*

It is used to set up, maintain, and inspect the Ethernet bridge configuration in the linux kernel. An Ethernet bridge is a device commonly used to connect different Ethernet networks together, so that these Ethernets will appear as one Ethernet to the participants. Each of the Ethernets being connected corresponds to one physical network interface in the bridge. These individual Ethernets are bundled into one bigger ('logical') Ethernet, this bigger Ethernet corresponds to the bridge network interface. In Figure 8, we can see that *panbr* is a bridge which include both network interfaces eth0 and ra0.

```
root@joao-desktop:/home/joao# brctl
Usage: brctl [commands]
commands:
        addbr           <bridge>                add bridge
        delbr           <bridge>                delete bridge
        addif           <bridge> <device>       add interface to bridge
        delif           <bridge> <device>       delete interface from bridge
        setageing       <bridge> <time>         set ageing time
        setbridgeprio   <bridge> <prio>         set bridge priority
        setfd           <bridge> <time>         set bridge forward delay
        sethello        <bridge> <time>         set hello time
        setmaxage       <bridge> <time>         set max message age
        setpathcost     <bridge> <port> <cost>  set path cost
        setportprio     <bridge> <port> <prio>  set port priority
        show                                    show a list of bridges
        showmacs        <bridge>                show a list of mac addrs
        showstp         <bridge>                show bridge stp info
        stp             <bridge> {on|off}       turn stp on/off
root@joao-desktop:/home/joao# brctl show
bridge name     bridge id               STP enabled     interfaces
panbr           8000.000c6e89bdc2       no              eth0
                                                        ra0
root@joao-desktop:/home/joao#
```

Figure 8 – *brctl* output

4.1.7 *route*

It manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the *ifconfig* program. When the 'add' or 'del' options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables. With this tool we can check if the computer is connected to the Internet or other external networks. In the case presented in Figure 9 the host device is connected to a local area network through interface nas0 and it is connected to the internet using the ppp0 interface which is the default route for IP connections.

```
root@joao-desktop:/home/joao# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
212.0.167.185   *               255.255.255.255 UH    0      0        0 ppp0
192.168.0.0     *               255.255.255.0   U     0      0        0 nas0
default         *               0.0.0.0         U     0      0        0 ppp0
root@joao-desktop:/home/joao#
```

Figure 9 – *route* output

4.1.8 *iptables*

It is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains. Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called a `target', which may be a jump to a user-defined chain in the same table.

4.1.9 *grep*

This command searches the named input files for lines containing a match for the given patterns. Matching lines are printed by default. The standard input is searched if no files are given or when the file - is specified. This tool was used to retrieve information from *ifconfig*'s, *iwconfig*'s and *iwlist*'s outputs. Next is presented an output example of the *grep* command used after an *ifconfig* call where only lines with the "nas0" word are presented.

```
root@joao-desktop:/home/joao# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:272 (272.0 b)  TX bytes:272 (272.0 b)

nas0      Link encap:Ethernet  HWaddr 00:0E:50:6D:3A:69
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:50ff:fe6d:3a69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2086 errors:0 dropped:2 overruns:0 frame:0
          TX packets:1133 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2788677 (2.6 MiB)  TX bytes:107741 (105.2 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:87.196.24.208  P-t-P:212.0.167.185  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1492  Metric:1
          RX packets:1952 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1075 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:2763870 (2.6 MiB)  TX bytes:70582 (68.9 KiB)

root@joao-desktop:/home/joao# ifconfig | grep nas0
nas0      Link encap:Ethernet  HWaddr 00:0E:50:6D:3A:69
root@joao-desktop:/home/joao#
```

Figure 10 – Output of *grep* used after *ifconfig*

4.1.10 *awk*

It scans each input file for lines that match any of a set of specified patterns. With each pattern there can be an associated action that will be performed when a line of a file matches the pattern. Each line is matched against the pattern portion of every pattern-action statement; the associated action is performed for each matched pattern. Using this program with *grep* we were able to retrieve only one text field from a program output. This procedure will be explained in detail in Chapter 5. Figure 11 shows how we can retrieve an interface MAC address combining the *ifconfig*, *grep* and *awk* programs.

```
root@joao-desktop:/home/joao# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:272 (272.0 b)  TX bytes:272 (272.0 b)

nas0      Link encap:Ethernet  HWaddr 00:0E:50:6D:3A:69
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:50ff:fe6d:3a69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2090 errors:0 dropped:2 overruns:0 frame:0
          TX packets:1136 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2788861 (2.6 MiB)  TX bytes:107869 (105.3 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:87.196.24.208  P-t-P:212.0.167.185  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1492  Metric:1
          RX packets:1952 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1075 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:2763870 (2.6 MiB)  TX bytes:70582 (68.9 KiB)

root@joao-desktop:/home/joao# ifconfig | grep nas0
nas0      Link encap:Ethernet  HWaddr 00:0E:50:6D:3A:69
root@joao-desktop:/home/joao# ifconfig | grep nas0 | awk '{print$5}'
00:0E:50:6D:3A:69
root@joao-desktop:/home/joao#
```

Figure 11 – Output of *awk* used after *ifconfig* and *grep*

4.1.11 *dhcpd3*

The Internet Systems Consortium DHCP Server, *dhcpd3*, implements the Dynamic Host Configuration Protocol (DHCP). DHCP allows hosts on a TCP/IP network to request and be assigned IP addresses, and also to discover information about the network to which they are attached.

The DHCP protocol allows a host which is unknown to the network administrator to be automatically assigned a new IP address out of a pool of IP addresses for its network. In order for this to work, the network administrator allocates address pools in each subnet and enters them into configuration file. On startup, *dhcpd3* reads the configuration file and stores a list of available addresses on each subnet in memory. When a client requests an address using the DHCP protocol, *dhcpd3* allocates an address for it. Each client is assigned a lease, which expires after an amount of time chosen by the administrator (by default, one day).

The PAN Master runs this daemon in order to set up the IP network over the PAN. This procedure will be explained in Chapter 5.

```
root@joao-desktop:/home/joao# dhcpd3 eth0
Internet Systems Consortium DHCP Server V3.0.3
Copyright 2004-2005 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Wrote 1 leases to leases file.
Listening on LPF/eth0/00:0c:6e:89:bd:c2/192.168.1/24
Sending on   LPF/eth0/00:0c:6e:89:bd:c2/192.168.1/24
Sending on   Socket/fallback/fallback-net
root@joao-desktop:/home/joao#
```

Figure 12 – *dhcpd3* output

4.1.12 *dhclient*

The Internet Systems Consortium DHCP Client, *dhclient*, provides a means for configuring one or more network interfaces using the Dynamic Host Configuration Protocol. The DHCP protocol allows a host to contact a central server which maintains a list of IP addresses which may be assigned on one or more subnets. A DHCP client may request an address from this pool, and then use it on a temporary basis for communication on network. The DHCP protocol also provides a mechanism whereby a client can learn important details about the network to which it is attached, such as the location of a default router, the location of a name server, and so on.

```
root@joao-desktop:/home/joao# dhclient eth0
Internet Software Consortium DHCP Client 2.0pl5
Copyright 1995, 1996, 1997, 1998, 1999 The Internet Software Consortium.
All rights reserved.

Please contribute if you find this software useful.
For info, please visit http://www.isc.org/dhcp-contrib.html

Listening on LPF/eth0/00:0c:6e:89:bd:c2
Sending on   LPF/eth0/00:0c:6e:89:bd:c2
Sending on   Socket/fallback/fallback-net
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
DHCPOFFER from 192.168.1.1
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.1
bound to 192.168.1.3 -- renewal in 1296000 seconds.
root@joao-desktop:/home/joao#
```

Figure 13 – *dhclient* output

In the output of the *dhclient* presented in Figure 13 the four messages of the DHCP Protocol as mentioned in Chapter 3.3.

4.1.13 *pand*

The PAN daemon [17] allows us to connect to Ethernet networks using Bluetooth. This deamon creates a virtual bnepX interface which is directly related to a hciX Bluetooth interface. This new bnepX interface encapsulates Ethernet frames in a Logical Link Control and Adaptation Protocol (L2CAP) session. PAN is the Bluetooth profile using BNEP and defining how to do networking. It includes the definition of the Service Discovery Protocol (SDP) attributes and three node roles. In a typical Ad-Hoc scenario, the server will have the role GN (Group Ad-Hoc Network), and up to 7 clients with role PANU (Pan User) can be connected to it. An Access Point would take a NAP (Network Access Point) role.

4.2 Other Software Tools Used

4.2.1 Ethereal/Wireshark

Ethereal/Wireshark is the standard network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. We could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course). Ethereal was mainly used to troubleshoot network problems and to debug the protocol implementation. With it we could determine if PCP messages were correctly sent and/or received. This was very helpful and fast way to debug the software.

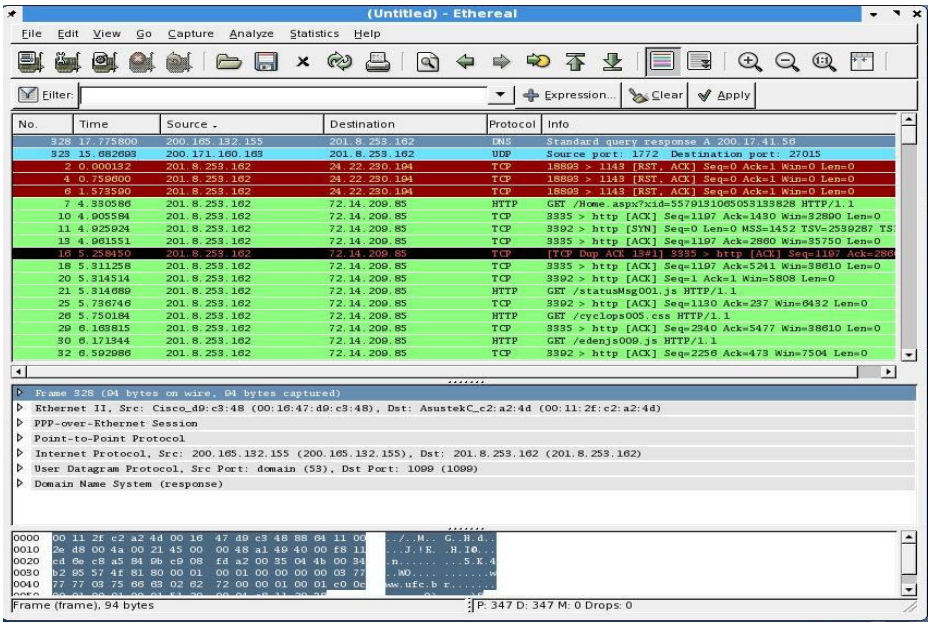

Figure 14 – Ethereal GUI

4.2.2 PackETH

PackETH is a Linux GUI packet generator tool for Ethernet. It allowed creating and sending any possible packet or sequence of packets over Ethernet. With it we were able to create our custom protocol messages and send them through Ethernet. This way, only if the message passed the test, we would add it to our software. We just had to fill in the boxes with the destination and source MAC addresses, the protocol type and the body of the message and press the Send button. If the message was well parsed by our software, then we could implement it. Together with the help of Ethereal we tested the PCP protocol.

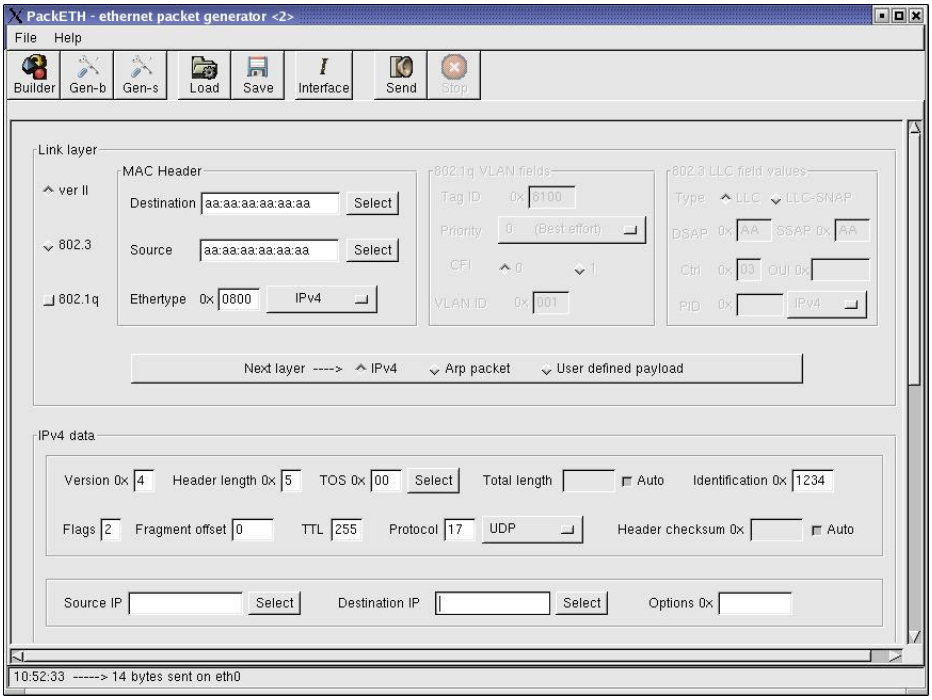

Figure 15 – PackETH GUI

4.2.3 Iperf

Iperf is a commonly used software network testing tool that can create TCP and UDP data streams to measure the throughput of a network that is carrying them. The output of the program is a text file where the results are presented in form of a table allowing the user to easily port it to other formats to do a more thorough analysis. This program was used in the testing phase of the project.

Having presented the software tools used along the work and the background theory in the previous chapter the work description will be detailed in the next chapter.

5 Work Description

In this chapter the various stages of the implementation will be described. The current implementation considers a subset of the features specified by ASPAN. It represents a proof-of-concept ASPAN prototype that opens the door to future enhancements according to the ASPAN specification. The main technologies enclosed by the prototype are Bluetooth, Wi-Fi and Ethernet (cable). Using different technologies it was possible to test the viability of ASPAN in heterogeneous networks. Bluetooth and Wi-Fi were the obvious wireless choices because they are the ones with better support and the most implementation in the market. Also, the standard Ethernet cable link was also implemented to increase the number of different technologies used. Although not a wireless technology, its coexistence with other communication technologies within a future PAN is a strong possibility. The implementation was made in C/C++ over Linux OS and it was divided in several stages which are presented in the following sections.

5.1 Interface Finder Module

In this section, it is explained how the available network interfaces are found. For each interface available a structure named "interface" (see Annex B) is created and filled with its characteristics and settings. In the Management State Machine Module (see section 5.1.3) every interface will be handled through this structure. This allows us to make the treatment of data independent from interface type.

Using the Linux built-in programs we discover both Bluetooth and Wi-Fi interfaces. Before making them available for PAN use we check if they are already being used for other purposes, such as to access to the Internet. In order to do this, the IP routing table is scanned for default routes. If such route is found the user is asked to choose between the external access and allowing the interface to be used for internal network (PAN). This is the only time the user may be asked to decide something in the PAN. From now on, the process is totally automated. If the user decides to use this interface for external access the process setup for this interface is not initiated and this interface remains external to the PAN. Else if the user decides to use the interface for the internal network it is removed from the routing table and the interface structure is filled with the proper values.

5.2 Interface Configuration Module

This section explains how the network interfaces available for PAN use were configured. Since the configuration is quite different for the different wireless technologies, the description for each technology is provided in separate sections.

5.2.1 Bluetooth Interfaces

Firstly the Bluetooth device name is set as devicename.PANname as explained in Section 3.2.3 (e.g., laptop.BobsPAN). Afterwards, the device searches for possible PAN

neighbours. This is done by scanning the neighbourhood, using Linux Bluetooth proper tools, for other devices with the same PANname. If no device is found, the interface stays in stand-by mode, waiting for the arrival of PAN connections. Otherwise, if a device is found with a name ending in PANname, its MAC address is saved and we connect to it creating a level 2 link between both devices. This is done using the Linux's pand (pan daemon). This interface is now flagged to broadcast a PCP ELECTION message.

5.2.2 WLAN Interfaces

In this case, as referred in Chapter 3.2, the SSID identifies the WLAN network and is used to physically distinguish between co-located WLAN networks. Thereby, we need to be careful how we name our WLAN network interfaces. In the beginning, we scan for WLAN networks with the SSID ending in "PANname". After that we have three ways to continue, depending on the scanning results:

1. If no network with PANname is found we set the SSID to devicename.PANname and configure the network interface to listen for the arrival of new devices to form the PAN.
2. If a network named other-devicename.PANname is found, it means that there is someone already waiting to form the PAN and we can connect to it. So we change our devicename to other-devicename.PANname to allow communication and initiate the master election process, broadcasting the Election message.
3. If a network named PANname is found, it means that the PAN is already established and so we have to proceed with the "Join" mechanism. We change our name to PANname to enable communication.

At this point there is a link established between both Wi-Fi interfaces allowing the exchange of PCP messages.

5.3 Running External Software Programs

As mentioned in Chapter 4 some external programs are used in order to obtain system information, to connect to other devices and retrieve device status. These programs were run the prototype using the *exec()* family of functions. Some of these programs needed for input the output of other programs, namely *grep* and *awk* to retrieve predetermined fields of an output (see Sections 4.1.9 and 4.1.10). This was implemented with *pipe()* and *waitpid()* functions allowing a fast execution of the external programs. This was very important because the prototype often needs to run these programs and their execution time is directly reflected in the prototype's performance.

5.4 The *wire* Structure

In order to compute the proper active topology tree using the spanning tree algorithm, defined by the ASPAN framework, the PAN Master must have information about all the existing links within the PAN. This information is stored in a data structure called *wire*. The

wire structure has 8 fields that fully characterize each link: first name, first interface, first mac, second name, second interface, second mac, link type and active. First and second name identifies the pair of devices connected to the link, first and second interface have the name of the network interfaces in those devices interfacing with the link, and first and second mac have the corresponding MAC addresses of the network interfaces. Obviously, "first" and "second" distinguish the two ends of the link. The link type field at this stage can be one of the three: Bluetooth, Wi-Fi or Ethernet. The active field is a Boolean variable which is set by the output of the spanning tree algorithm. The prototype has several other variables and structures but this structure takes special relevance because the entire action behind the prototype dynamic evolves around the values stored in them.

5.5 The Sub Master role

The Sub Master is a passive member in the PAN like all other peers with exception for the PAN Master. None of the decisions are performed by it. The Sub Master acts only as a backup for the Master. All the information the Master needs to manage the PAN is also stored in the Sub Master. If the Master leaves the PAN the Sub Master is promoted to Master. Since it has all the information needed to manage the PAN no harm is cause by the leaving of the Master.

5.6 Management State Machine Module

In the running state of the PAN there is a constant need of checking its status and its changes in order to manage the network and keep it working. So, after the network interfaces are found out and properly configured we move on to a state machine. This state machine works according to one major parameter: the type of message received. According to the message received and the current status of the network the prototype will perform a number of predetermined tasks and then return to the "wait for message" state. This can be viewed as a first-come first-served service policy where the first message to arrive is handled first and the second waits until the first one is finished, etc.

This management state machine included features such as handling the PAN Master election mechanism, discovering the network topology, setting up IPv4 connectivity and bridge configuration. For a better understanding they will be explained in separate sections.

5.7 PAN Master Election Mechanism

The first step for creating the PAN is to elect a PAN Master. In this particular step of the PAN setup the implemented solution is different from the one stated in the framework (Chapter 3.1.2.2). As shown on Figure 3 according to the ASPAN device #2 starts the election process sending an Election message to devices #1 and #3 and they forward the message to their neighbours (if any). In terms of software implementation one thing is given: every device run the same software therefore "think" the same way. Then, one may ask: why

would device #2 be the one to start the election process and not any other one? To solve this problem a new simpler election mechanism was implemented.

The election process only involves the first two devices arriving to form a PAN. The first device to arrive will not find any other PAN member so it will remain in a "wait election" state until other device arrives. As the second device scans for the PAN it will find device one and starts the election process. The election process includes two messages as shown in Figure 16.

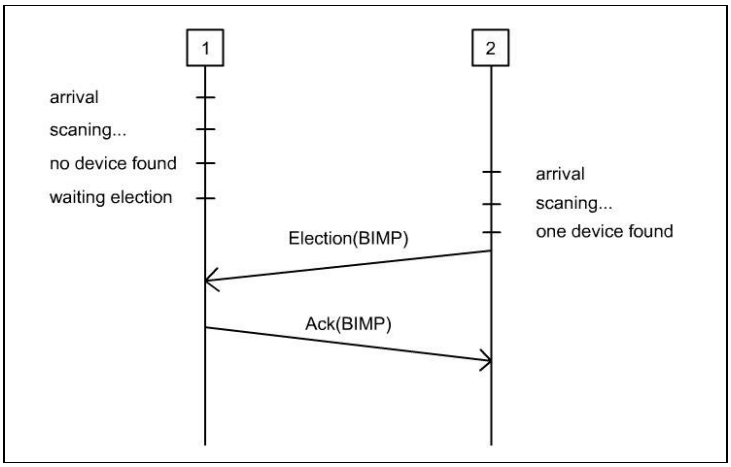

Figure 16 - Election process

Both these messages contain the same information about the sender (see Annex A). This can then be seen as a decision rather than an election. Each peer receives the information about the other and decides if it is going to be Master or Sub Master. There is no need for one to inform the other about which role to take. Also, it is important to realize that with the information provided by the received message the Master can already create the first *wire*, since all the information needed for filling it in is available.

5.8 Brigde Configuration and IPv4 Network Setup

The PAN device to communicate with its peer has to do it through a network interface. The ASPAN solution assumes devices with more than one communication technology. In order to have more than one on interface working in the same network they are configured as bridges. By default, every device creates a bridge either in the election process or in the join process. This bridge will be referred as *panbr* from now on. For instance, device Laptop has two interfaces: W which is a Wi-Fi interface and B which is a Bluetooth interface. During the initial process Laptop discovers Desktop device through a Bluetooth scanning and discovers no peers with the Wi-Fi scan. Laptop's Bluetooth interface is flagged to send a join/election message. Before the election/join message is sent, Laptop adds interface B to the *panbr* bridge. All the active PAN interfaces will be in the *panbr* while the ones which are not active (and not active does not mean they don't have any connection established) will not. Whether a network interface is added to the bridge or not depends on

the active topology computed by the master using the spanning tree algorithm considered in the ASPAN framework.

Structuring bridges as explained before allows an easier deployment of an IPv4 networking (or any other networking protocol). The PAN Master runs the DHCP Server and all other peers run the DHCP client. The entire IP configuration will be performed in the *panbr* interface of all and each one of the PAN peers. With this, the PAN dynamics due to the arrival and/or departure of devices becomes transparent to the IP layer and the layers above. As new links are created and some are terminated their respective interfaces are simply added or removed to/from the panbr.

5.9 Network Topology Discovery and Spanning Tree Algorithm

The spanning tree algorithm ran by the PAN Master is at the core of the management process of the PAN. The information stored in the wire structures is used as input to the algorithm. The spanning tree algorithm returns the links that should be active in order to define the active topology of the PAN. The Master then translates that information into PAN language, that is, which interfaces must be in the bridge (*panbr*) of each individual PAN device, comparing it with the *wire* structures. After applying the necessary changes to its own bridge the Master broadcasts a BRIDGE message is towards all members so that they can do the same. Each peer applies its new bridge configuration before forwarding the message. This ensures that there is no message loop in the process and each peer will only receive one BRIDGE message since it is always forward through the new defined path.

5.10 Managing the Arrival of New Devices to the Network - Join

After the PAN is setup, new devices can join it. From the arriving device's point of view the process is similar to the election process. First it scans for peers and once it found them the election process takes place as usual. The difference has to do with the reply to the election/join message. The PAN peer upon receiving the join/election message fills a *wire* structure characterizing the new link and sends a NEW_DEVICE message to the PAN Master with this information. Meanwhile the arriving device is still on hold for an answer. The PAN Master then answers to the new arriving device with one of the three possible messages according to the new device's BIMP:

- UR_MASTER - The arriving device is the new PAN Master. This message includes all the network topology information, including the new link so that the new master can execute the spanning tree algorithm to decide which links should be active or not. Also the PAN Master demotes itself to Sub Master and sends a S_TO_P message to the Sub Master demoting it to Peer. The demoted Sub Master erases all the information it had in memory regarding the PAN topology.
- UR_SUBMASTER - The arriving device is the new PAN Sub Master. This message also includes the network topology since the Sub Master always has as

copy of the PAN network. Again a S_TO_P message is sent to the Sub Master do demote it to Peer.

- UR_PEER - The arriving device is an ordinary PAN Peer. A TREE message is sent to the Sub Master with the new PAN topology tree information.

If any of the two last to messages are sent, the Master remains the same so the next step is to execute the spanning tree algorithm. If not then the previous Master device concludes its role in the joining process and wait in a passive state for Master's messages. Every time a peer is promoted to Master it broadcasts an ANNOUNCE message to the entire PAN. This message will inform all peers which device is Master and where it is in the PAN.

5.11 Managing the Arrival of Legacy Devices to the Network - Guest

The guest mechanism is an alternative to the join mechanism presented in the section above. In this case, the arriving device is not running the ASPAN prototype. Therefore no PCP messages can be exchanged between the guest device and the other PAN devices. Nevertheless, since the PAN is using a standard auto configuration mechanism the guest can use that same mechanism to take advantage of some of the PAN features. In this prototype we use DHCP and the PAN Master is running a DHCP Server, so if the guest device runs the DHCP client it will acquire a PAN IP address enabling it to communicate with all other peers. Moreover, if the PAN is connected to an external network such as the Internet, the guest device will also have access to it. The downside of not having the ASPAN prototype is that the entire configuration in the guest device must be done by the user.

5.12 Managing the Departure of PAN devices

To finalize the explanation of the mechanisms dealing with the PAN dynamics, the departure of devices has to be considered. Here, the Master removes one or more wire data structures to its local list depending on the number of links associated to the leaving device. The leaving device informs its neighbours it is leaving with a LEAVE message. But, like in the join mechanism, here there are also some different possibilities:

- Master is the leaving device: in this situation before the Master sends the LEAVE message it sends a S_TO_M message to the Sub Master; the Sub Master is promoted and then when the LEAVE message is sent it is forward to the new Master; afterwards, a new Sub Master is elected.
- Sub Master is the leaving device: the LEAVE message is forward to the Master and then a new Sub Master is elected.
- Peer is the leaving device: the LEAVE message is forward to the Master; no further actions are needed.

Upon receiving the LEAVE message, the neighbours immediately make the interface which was previously connected to the leaving device available for new incoming join

requests from possible new PAN devices and forward that LEAVE message to the PAN Master. After receiving the LEAVE message(s) the Master deletes all wires containing the leaving device's name (see Section 5.1.5) and executes the spanning tree algorithm to compute the new active PAN topology.

In this chapter the work was presented in detail. It describes all the major features and how they were implemented. The initial device setup was presented where each device configures its interfaces so they can communicate with other devices. Next the PAN configurations were referred. A device is elected as Master and its role within the PAN was defined. The Master manages the PAN by setting up its active topology and handling the dynamics associated to it. The arrival and departure of devices and its implementation was also referred. This concludes the work description.

6 Work Evaluations

In this chapter we refer to a set of tests performed in order to evaluate the ASPAN prototype developed along this work. A description of the tests, the scenarios used to perform them, and the obtained results are presented. The two major goals of these evaluations were: 1) to understand whether the prototype met the ASPAN requirements; 2) verify whether the configuration times were acceptable from the user point of view.

6.1 Setup and Election Process

In order to test the election process two devices are used. Device #1 starts first. As it does not find any other devices it remains in a non-PAN state waiting for other devices. Then device #2 starts. It scans for other devices, finds device #1, connects to it and starts the election process. The figures below represent two PAN setups used in this test.

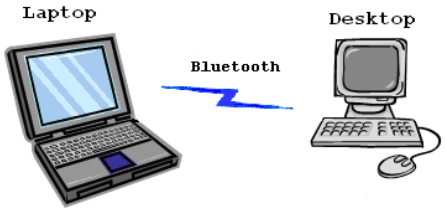

Figure 17 - Bluetooth PAN setup

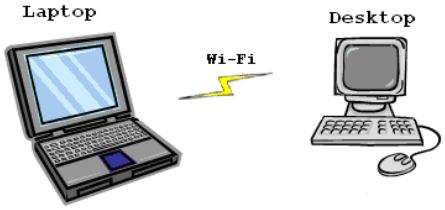

Figure 18 - Wi-Fi PAN setup

Every PAN creation begins with this process even if there are more than two devices to initially form it. In the following figures we present an example of the prototype's output in both devices when a PAN is being created using Bluetooth as the underlying communication technology. The device #1 is named as "laptop" and device #2 as "desktop". The clocks of the two devices were synchronized to provide a better reading of the messaging events between them.

```
(20:01:05) ## PAN name: PAN
(20:01:05) ## Scanning for Bluetooth PAN member(s)...
(20:01:16) ## No member was found!
(20:01:16) ## Number of PAN interfaces: 2
(20:01:16) ## bnep0: WAIT_ELECTION   |  eth0: WAIT_ELECTION

--> MESSAGE TYPE [1] RECEIVED
(20:01:36) ## Election Received from desktop at bnep0
(20:01:36) ## Ack sent through bnep0
(20:01:36) ## Bridge created [bnep0]
(20:01:37) ## DHCP Server Configured
(20:01:37) ## :: Topological tree ::
(20:01:37) ##  [laptop:bnep0]<------BT------>[desktop:bnep0]    (enabled)
(20:01:37) ## -----------------------------------------------------
(20:01:37) ## desktop bridge = [ bnep0 ]
(20:01:37) ## Bridge setup sent to PAN members
(20:01:37) ## ANNOUNCE sent through bnep0
(20:01:37) ## RANK: MASTER
(20:01:37) ## bnep0: STANDBY   |  eth0: WAIT ELECTION
```

Figure 19 - Output of device #1

```
(20:01:23) ## PAN name: PAN
(20:01:23) ## Scanning for Bluetooth PAN member(s)...
(20:01:33) ## 1 member was found! MAC:00:80:5A:30:87:78
(20:01:35) ## waiting for bnep0 to get up...
(20:01:35) ## waiting for bnep0 to get up...
(20:01:36) ## Number of PAN interfaces: 2
(20:01:36) ## bnep0: SEND_ELECTION   |  eth0: WAIT_ELECTION
(20:01:36) ## Election sent through bnep0
(20:01:36) ## bnep0: WAIT_ACK   |  eth0: WAIT_ELECTION

--> MESSAGE TYPE [2] RECEIVED
(20:01:36) ## Ack Received from laptop at bnep0
(20:01:36) ## Bridge created [bnep0]
(20:01:36) ## RANK: SUBMASTER
(20:01:36) ## bnep0: STANDBY   |  eth0: WAIT_ELECTION

--> MESSAGE TYPE [14] RECEIVED
(20:01:37) ## Bridge setup [ bnep0 ]

--> MESSAGE TYPE [13] RECEIVED
master is laptop@bnep0 with bimp 980.000000
(20:01:37) ## dhclient started.
(20:01:39) ## PAN IP: 10.0.0.200
```

Figure 20 - Output of device #2

The time values shown in Figure 21 were measured at device #2 from the start of the execution until an IP address is acquired. The procedure was repeated 10 times considering either Wi-Fi or Bluetooth as the underlying technology.

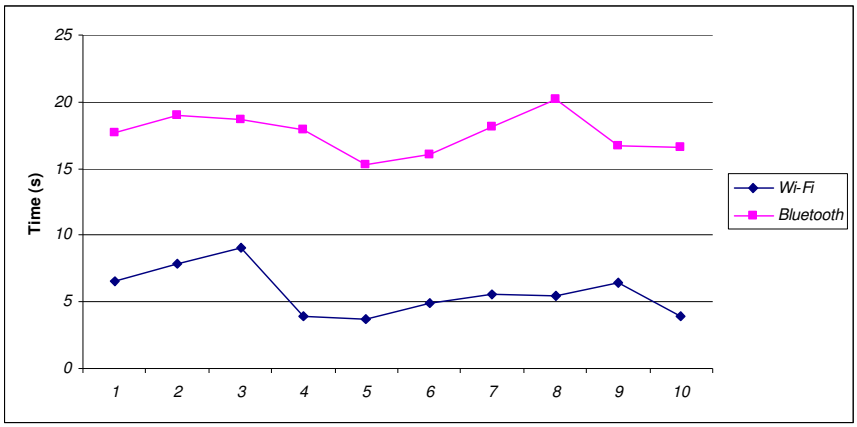

Figure 21 - PAN setup time either with a Wi-Fi or Bluetooth link

With a 95% confidence interval, the expected value for setup time for Bluetooth link setup is in the interval [16.7s , 18.5s] and for the Wi-Fi is in the interval [4.6s , 6.8s]. The main reason for such a substantial difference is the time that the Bluetooth scan takes to return its neighbour devices. This test was made in a home/noise free environment. Other tests performed in more crowded areas, where a considerable number of Bluetooth devices could be found after a scanning process, revealed much higher setup times; the setup time could take up to a minute. Although the scanning time is included here it is something we can not control/improve in our prototype.

6.2 Arrival of a new PAN Master

In this section the test for the worst join case scenario in presented. It is the worst scenario because in this case the new arriving peer will be elected Master and its process includes more message exchange than any other. Also the DHCP Server has to stop in the previous master and be started in the new elected Master. The scenario used for this test initially considers a PAN formed by 3 devices connected as depicted in Figure 22.

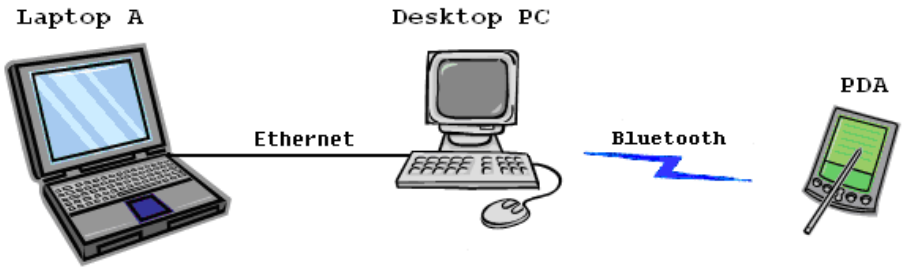

Figure 22 - Initial PAN setup

At this point we have an established working PAN. "Laptop A" is the Master, "Desktop PC" is Sub Master and each PAN device has an IP address assigned. Next, a fourth device named "Laptop B" with a Wi-Fi interface arrives and connects to the

"Desktop PC" which also has a Wi-Fi interface available for PAN use. Due to its characteristics "Laptop B" is elected PAN Master. The final PAN topology is depicted in Figure 23.

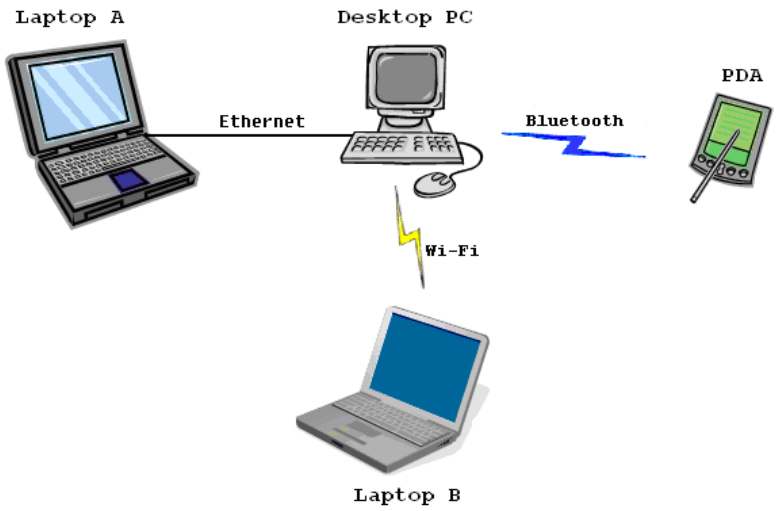

Figure 23 - Final PAN setup

The following figures show the reconfiguration time for "Laptop A" and "Desktop PC", respectively.

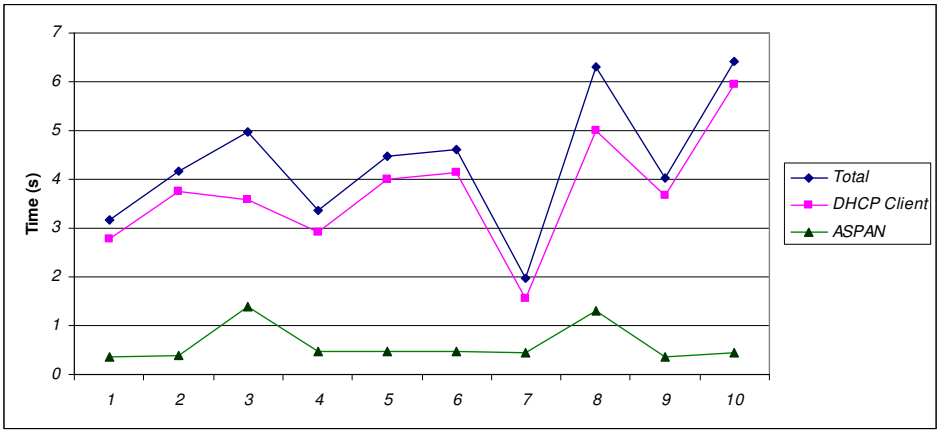

Figure 24 - Reconfiguration time for "Laptop A" when demoted to Sub Master

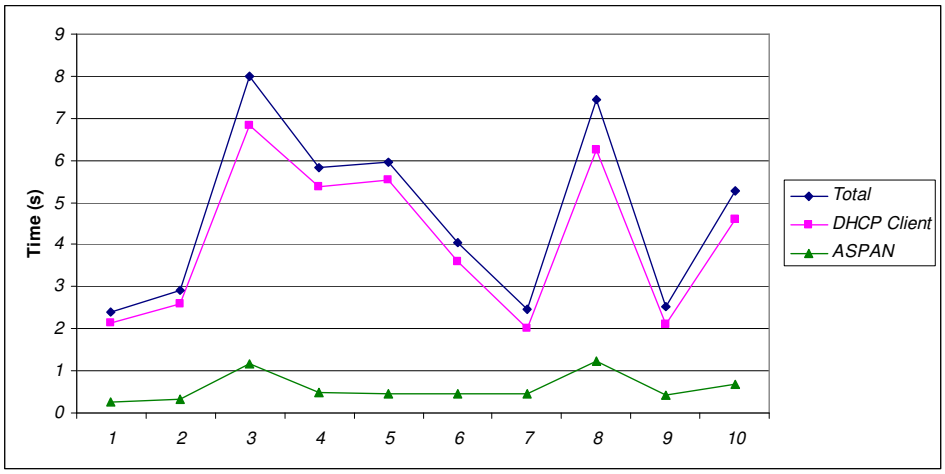

Figure 25 - Reconfiguration time for "Desktop PC" when demoted from Sub Master to Peer

Two steps are needed for the reconfiguration: first the elected Master has to compute the active topology tree and communicate to the other members their respective bridge configurations. Afterwards, the Master broadcasts a DHCLIENT message and only then the remaining peers run the DHCP Client to acquire an IP address.

In average, the total reconfiguration time for "Laptop A" is 4,3 s. However, the ASPAN process only takes about 14% of the total reconfiguration time (0,6 s) to complete; the rest of the reconfiguration time (3,7 s) is due to the auto-configuration mechanism (DHCP). For the "Desktop PC", the total reconfiguration time is 4,7 s. In this case, the ASPAN process takes 13% of the total reconfiguration time (0,6 s) to complete. The remaining 4,1 s are used by the DHCP. The configuration time for "Laptop B", the arriving new Master, was around 6 s as stated in the previous section. Although this arriving device needs extra time for the Wi-Fi scanning process, once elected as the new PAN Master, it only has to setup a DCHP Server to finish its setup. As such, the extra time spent on scanning is compensated by not running the DHCP Client.

6.3 Departure of the PAN Master

In contrast to the scenario presented in the previous section, here the worst leave case scenario will be evaluated. The initial PAN configuration for this test is the same as the final PAN configuration in the previous section and is shown in Figure 26.

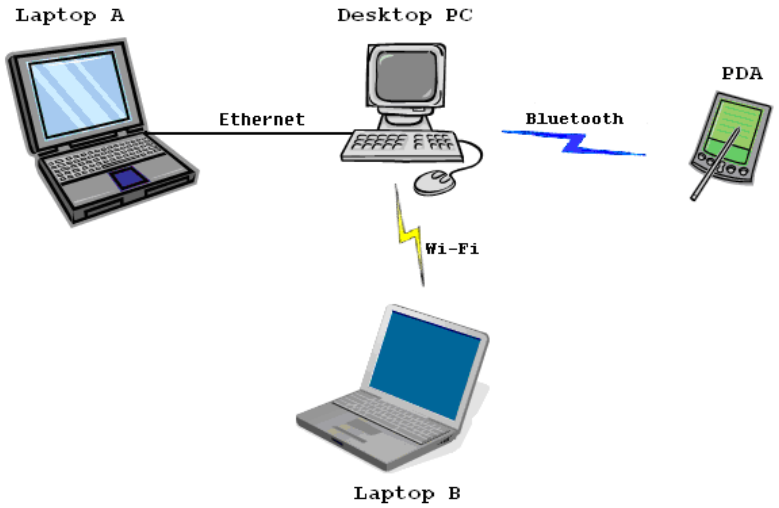

Figure 26 - Initial PAN setup

Laptop B" is the PAN Master and "Laptop A" is the Sub Master. When the Master leaves the Sub Master will be forced to take its place, compute the new active topology tree, and instruct the other peers to act accordingly. After the reconfiguration the devices will be connected to each other as presented in Figure 27 having "Desktop PC" figuring as the new elected Sub Master.

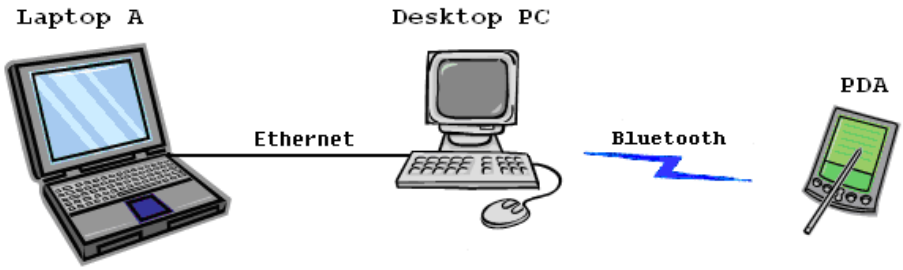

Figure 27 - Final PAN setup

The procedure was repeated 10 times. The reconfiguration times for "Laptop A" and "Desktop PC" are presented in Figure 28 and Figure 29.

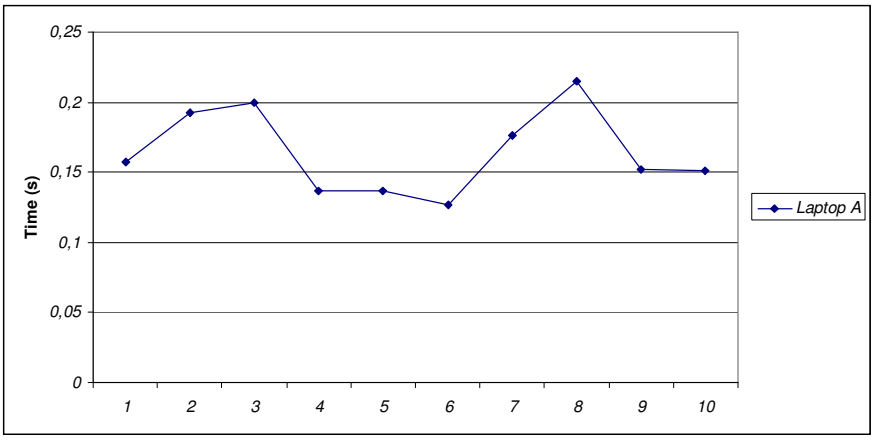

Figure 28 - Reconfiguration time for "Laptop A"

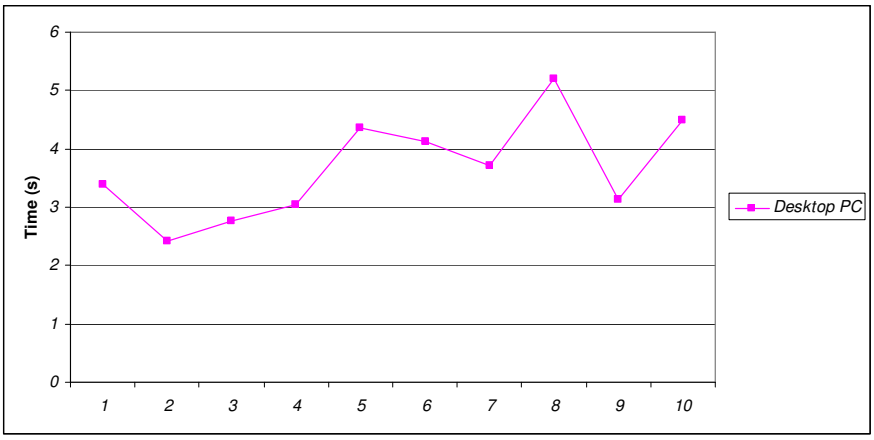

Figure 29 - Reconfiguration time for "Desktop PC"

The average reconfiguration time for a PAN Sub Master when the Master leaves the PAN is less than 170 ms. Within this time the "Laptop A" is notified about the departure of "Laptop B", computes the new active topology tree, notifies its peers about their new setup, and configures and starts a DHCP Server. The reconfiguration time for the "Desktop PC" it is quite longer because it has to run the DHCP Client which, as seen in Figure 24 and Figure 25, takes about 4 seconds to complete.

6.4 Data Exchange between PAN Peers

In this section the exchange data rate obtained between PAN peers is presented. This test demonstrates the effect of using bridges in this prototype. For that purpose, the scenario illustrated in Figure 30 was considered.

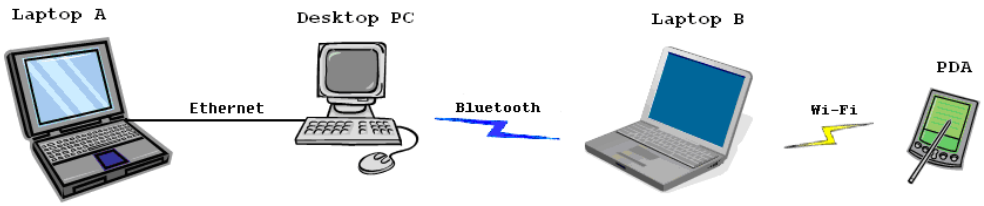

Figure 30 - Initial PAN Setup

In this PAN we have 3 distinctive types of links:

- Ethernet cable operating at 100 Megabits/12,5 Megabytes per second
- Wi-Fi link operating at 11 Megabits/1,37 Megabytes per second
- Bluetooth link operating at 723 Kilobits/90 Kilobytes per second (see section 2.1)

This scenario was chosen in order to create a heterogeneous PAN with two devices acting as bridges, "Desktop PC" and "Laptop B". The average data rates obtained between PDA and each one of the other three PAN devices is shown in Figure 31 and Figure 32.

During each test there was only one active data flow on the PAN between two peers. The remaining peers are in an idle state not consuming bandwidth.

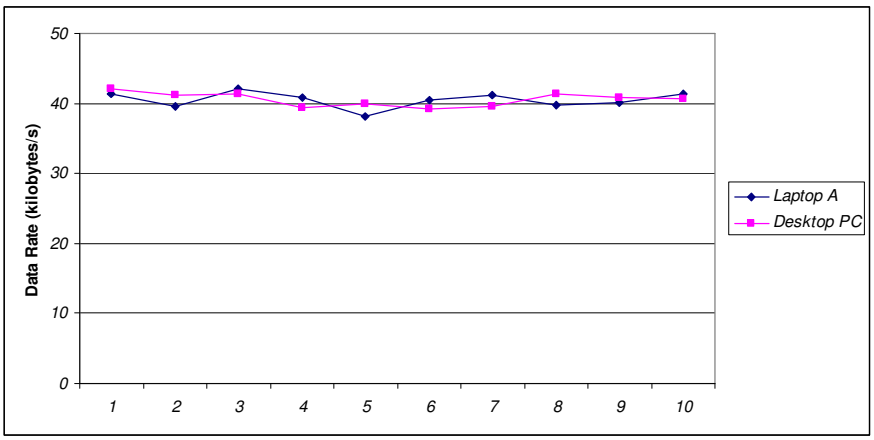

Figure 31 - Data rates between PDA-Laptop A and PDA-Desktop PC

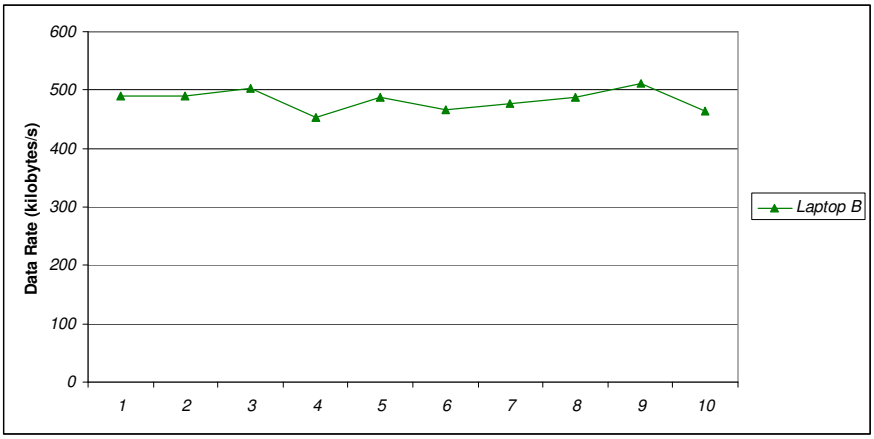

Figure 32 - Data rates between PDA-Laptop B

According to these values it is clear the Bluetooth link is the bottleneck in this network. Therefore two separated graphs were made to present the results; in there is a Bluetooth link between both presented devices while in there is not.

According to these values it is clear the Bluetooth link is the bottleneck in this network. Therefore, two separated graphs were made to present the results; in Figure 31 there is a Bluetooth link between both presented devices while in Figure 32 there is not. The average time for the "PDA - Desktop PC" and "PDA - Laptop A" was 41 kB/s in both which confirms the bottleneck at the Bluetooth link. Once the information passes that link the Ethernet cable has enough bandwidth to transmit the data. For the "PDA - Laptop B" case there is a direct Wi-Fi link between them. The average data rate was 485 kB/s. Both these values, 41 kB/s and 485 kB/s are according to what would be expected. Although the Bluetooth specifies 90Kb/s and the Wi-Fi specifies 1370Kb/s as their data rates it is

important to notice that the *iperf* program only counts as data the information in the application layer leading to the difference between such theoretical values and the data rates obtained in practice. After the test a connection was made to a local FTP Server using a Wi-Fi link. A large file was downloaded (700 Megabytes) and the average download speed was 490 Kilobytes per second confirming the values obtained within the PAN.

6.5 Discussion

As seen in section 6.1 the PAN setup is totally dependent of the wireless technologies in use to creating the links between the devices. Also, the scanning process inherent to each technology contributes with a significant portion of time when considering the total setup time. In Figure 20 the output of a connecting Bluetooth device is shown and we can see that the scanning takes 10 s, the connection creation and setup takes 3 s and the PAN configuration takes 3 s. In that particularly example the scanning took around 60% of the overall time.

After creating a PAN we tested a join scenario where the arriving peer is going to be the new PAN Master. This is a worst join case scenario because it involves more messaging exchange between the PAN devices and, therefore, takes more time. Figure 24 and Figure 25 relate to reconfiguration times of PAN members upon the arrival of a new device. We can se that the total reconfiguration time is greatly due to the DHCP Client which has to be executed. Its execution takes more than 85% of the total time in both cases leaving the ASPAN with an absolute reconfiguration time of 0.6 s. A closer look into those two figures tells us that the ASPAN reconfiguration time of both devices is related, which is expected since the reconfiguration itself results of messaging exchange between the PAN peers. Figure 33 shows both "Laptop A" and "Desktop PC" reconfiguration times.

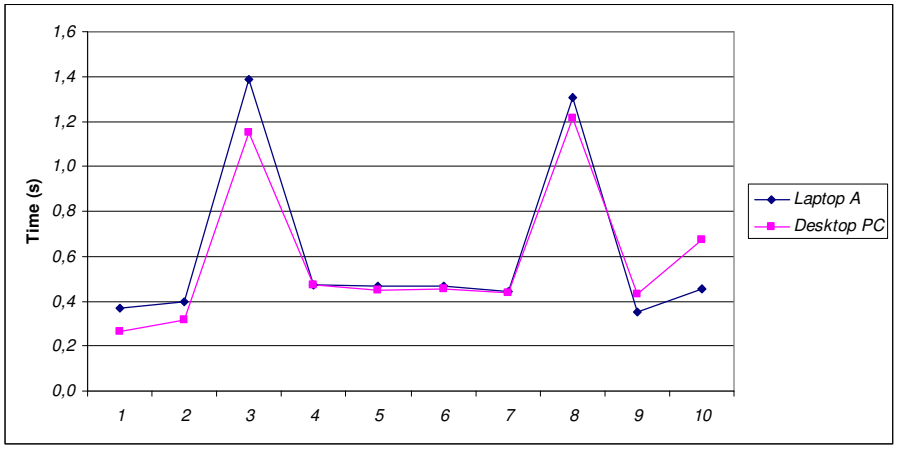

Figure 33 - Comparison between Prototype reconfiguration time of "Laptop A" and "Destkop PC"

Afterwards we tested a leave scenario being the Master the leaving device. This is a worst leave case scenario because the Sub Master has to configure and run a DCHP Server in order to enable the continuity of a Layer 3 network and to enable former new peers to

acquire IP addresses at their arrival. The average setup time for the new Master is around 160 ms while in the "Desktop PC", a regular peer, the reconfiguration time is around 4 s. Taking the example presented in Figure 33 we can say that the reconfiguration time for every PAN member is related. In this case, it is possible to say that the reconfiguration time in the "Desktop PC" is the same as the "Laptop A" added with the DHCP Client execution time. This shows that the reconfiguration time of a PAN depends on the self mechanism used to create a network over Layer 3, in our case the DHCP to create an IP network.

The final test was made to measure the network performance when data was exchanged over Layer 3. The results proved that the co-existence of the PCP over Layer 2 and an IP network established over Layer 3 has diminished influence in data rates. This was expected since the PAN uses few and very small messages (see Annex A).

Through these tests we verified that the ASPAN prototype works in accordance with the specification. Also the PAN set up time of approximately twenty seconds, is perfectly acceptable. The fast configuration and reconfiguration of the PAN when submitted to topology changes and the data rates obtained make this prototype a close representation of the solution defined by the ASPAN Framework.

Note: These tests were made with a final prototype. Beside the features mentioned in Chapter 5 this final prototype also includes some others made by Sérgio Lopes, regarding OSI Layer 3 aspects namely Internet connection properties which are also included in the ASPAN Framework. Therefore, one should notice that these results also include some additional configurations throughout the PAN setup process which are not part of this work.

7 Conclusions

With this work, we managed to conclude that the ASPAN framework and companion PCP protocol are a viable way of implementing the next generation PANs. Also, we realized that the use of bridges is an appropriate and easy solution for message forwarding within PANs, making the coexistence of multiple wired and wireless links within the same PAN completely transparent to IP and upper layers. The bridge based implementation allows us to create a single broadcast domain within the network which enables the direct use of standard protocols, such as DHCP. In addition, both wireless communication technologies used, Wi-Fi and Bluetooth, showed good compatibility when working together. This is directly related to the Bluetooth PAN profile and demonstrates that the latter is a valid option for next generation PAN setup. Moreover, the use of the DHCP protocol proved to be a good approach to setup an IPv4 network within a PAN. The overall prototype performance is directly related to wireless technologies used by the device in the PAN. As results show, during the bootstrap the configuration time depends on the communication technology used and the IPv4 network setup it is also dependent of the auto configuration mechanism used (DHCP). Therefore, the implemented solution only represents a small fraction of the total setup and/or configuration time.

In the following we recall the objectives of this work, point out the relevant results achieved, and mention the future work.

7.1 Work Revision

In this work we have created an ASPAN prototype as initially intended. To do that, some minor sub-objectives were defined and achieved. We started the work by developing a communication interface auto-configuration system which included the scanning for neighbour PAN members and the respective interface setup depending on the scan results and interface technology. Next, we enabled the communication between the different peers by implementing the PCP protocol over Layer 2. Using this protocol we were able to: 1) Employ the Master election process to define which peer would be in charge of managing the PAN; 2) Apply the best topology tree to the network, defined by the spanning tree algorithm, and consequent bridge configuration; 3) Use a standard self configuration mechanism such as DCHP to enable an IPv4 network over the previous established; 4) Set the PAN's dynamics allowing peers to join and leave the network.

7.2 Relevant Results

In this section, the main results obtained during this work are pointed out. For the sake of clearness, we use a different section for each result.

7.2.1 Next Generation ASPAN Prototype

The next generation ASPAN prototype represents our major result. To achieve this objective, we based our work on the solution developed by Campos and Ricardo. We implemented the PCP and some other configuration and management features. These are listed below in different sections.

7.2.1.1 Communication Interface Auto-configuration System

This system was created to eliminate user intervention during the PAN setup. The simultaneous presence of multiple communications technology and their individual characteristics turn this into a non-trivial process. Therefore, it was important to not involve the user in this task. The prototype supports Bluetooth, Wi-Fi and Ethernet technologies, which are configured automatically. Moreover, the user has no intervention during the whole PAN setup and managing phases. This was intended to enable the prototype to run as a background management service in the user's devices.

7.2.1.2 PAN Control Protocol Implementation

The PCP, implemented using Linux packet sockets, is the signalling protocol used to exchange control information between peers within the PAN and is independent from the underlying wireless/wired technologies used to create the links between them. The protocol uses an Ethernet frame to encapsulate its data and runs directly over Layer 2 (OSI model). This is essential because it allows communication between peers even before any kind of IP network is set up.

7.2.1.3 Master and Sub Master Election Mechanism Implementation

ASPAN is based on a master-slave model. Therefore an election mechanism had to be implemented to determine which device would be the PAN Master. This procedure is done in an early stage of the PAN creation and every time a new device joins it. The mechanism has two parts. Firstly, it collects information about the device's properties and according to an established algorithm classifies it. Secondly, it transmits the classification value to its peer. The device with the highest classification is elected. Also, a Sub Master is defined to prevent the network from being orphan if the Master leaves. This was implemented and proved to be a significant feature, as it made the network more reliable.

7.2.1.4 Spanning Tree Algorithm Implementation

The ASPAN uses a Minimum Routing Cost Tree algorithm [9] ran by the Master to determine the proper active topology tree for the network. With the algorithm's output the Master can successfully apply the resulting tree to the network. The changes are communicated to the peers using the PCP message BRIDGE (see Annex A) in order to enable/disable links. Having a tree as active network topology prevents loops and simplifies the forwarding process.

7.2.1.5 Creation of an IPv4 Network

After being elected the Master runs a DHCP Server to set up an IPv4 network. Afterwards, it instructs the remaining peers with the DHCLIENT PCP message to retrieve an IP address using the DHCP Client. In a few seconds an IP network will be established and each device will have its unique IP address enabling it to use any IP application. Features such as file transfer, audio streaming, and services sharing between PAN devices are now available.

7.2.1.6 Management of Arrival of New Devices to the Network

The process of arrival of new devices has two possible mechanisms: "join" and "guest". In the "join" mechanism, the new device announces its arrival to the network sending a PCP message to its neighbours. The neighbours, in turn, signal the Master about the new arriving device which adds it to the topology tree. The arriving device is then instructed to run the DHCP Client in order to obtain an IP address and be part of the existing PAN IP network. Also, special cases were implemented, namely the case where the arriving device has better capabilities than the PAN Master and, therefore, the topology information had to be transferred to the new Master. In the "guest" mechanism the joining device does not need to run the developed software nor to be running the same operating system. The new device only has to connect and ask for a PAN IP address using the predefined standard auto configuration mechanism, DHCP. However, this link needs to be secure to prevent "undesirable" guests. With this in mind, we just implemented the "guest" mechanism by cable (Ethernet) that obviously allows some control from the PAN user/owner. Nevertheless, considering that wireless technologies have built-in security features this kind of mechanism can also work wirelessly.

7.2.1.7 Management of Departure of New Devices to the Network

This procedure deals with the leaving of PAN devices. Before leaving the PAN, the leaving device informs its neighbours that it is leaving the network. In turn, the neighbours notify the PAN Master about that fact . This way the PAN Master can re-compute the new active topology tree (now excluding the leaving device) and notify the other PAN devices about the new setup. Also, the special case where the Master leaves the PAN was implemented. In this case, before leaving the Master informs the Sub Master so that the Sub Master can become the new Master and begin to manage the PAN.

7.3 Future Work

Taking the current prototype as basis, some new features can be implemented to enhance its performance and applicability. The following sections suggest some of the possible improvements that can be added.

7.3.1 Active Topology Check

To improve even further the dynamics of the network a topology monitoring mechanism could be implemented to handle the sudden departure of devices. The implementation of the mechanism already considered in the ASPAN framework to deal with this would be the next step. The ASPAN mechanism considers that the Master periodically inquiries the peers in order to know if all the links are still up. If some peer fails to answer or a link is proven to be down, the Master uses that information to re-compute and set a new topology tree to the network.

7.3.2 Security Features

At this stage, security is not taken into account in our PAN prototype. However, the intra-PAN connections could be encrypted using, for instance, the built-in security mechanism provided by the underlying wireless technologies used in our prototype.

7.3.3 Other Communication Technologies

Although this prototype is running with the two most used wireless technologies at the moment, Bluetooth and Wi-Fi, other communication technologies could be used in it. For instance, UWB could be naturally integrated in the current ASPAN prototype. In fact, the WiNet solution defined by the WiMedia Alliance [18,[19] is fully compatible with our prototype, as it defines a mean to transport Ethernet frames over UWB and specifies IEEE 802.1D bridges as the mean to interconnect UWB networks with other IEEE 802 networks. The integration of other technologies would be a way to improve the current prototype.

References

[1] - Bluetooth SIG, Specification of the Bluetooth System (version 2.0), November 2004.

[2] ANSI/IEEE Std 802.11, Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999 Edition (R2003)

[3] Standard ECMA-368, *High Rate Ultra Wideband PHY and MAC Standard*, December 2005

[4] Standard ECMA-369, *MAC-PHY Interface for ECMA-368*, December 2005

[5] M. Takizawa, et al., *MaCC: Supporting Network Formation and Routing in Wireless Personal Area Networks*, in Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04), March 2004

[6] R. Campos and M. Ricardo, *Autoconfiguration and Selfmanagement of Personal Area Networks: a New Framework*, in Proceedings of the 15th Meeting of the Wireless World Research Forum, December, 2005

[7] R. Wakikawa, et al. *Global connectivity for IPv6 Mobile Ad Hoc Networks,* Internet Draft, draft-wakikawa-manetglobalv6-05 (work in progress), March 2006

[8] R. Campos and M. Ricardo, *Dynamic and Automatic Connection of Personal Area Networks to the Global Internet*, in Proceedings of the ACM International Wireless Communications and Mobile Computing Conference (IWCMC'06) Wireless LANs and Wireless PANs (Wireless Networking), July 3-6, 2006.

[9] R. Campos and M. Ricardo, *A Fast Algorithm for Computing Minimum Routing Cost Spanning Tree,* submitted to Elsevier Computer Networks Journal.

[10] IEEE Workgroup 802.15.1, *Personal Area Networking (PAN) Profile*, June 2001

[11] Plummer, D., *An Ethernet Address Resolution Protocol - RFC-826*, November 1982

[12] T. Narten, E. Nordmark, W. Simpson, *Neighbor Discovery for IP Version 6 (IPv6)*, December 1998

[13] IEEE EtherType Field Registration Authority - http://standards.ieee.org/regauth/ /ethertype/index.shtml

[14] R. Droms, *Dynamic Host Configuration Protocol*, March 1997

[15] B. Croft, John Gilmore, *Bootstrap Protocol (BOOTP)*, September 1985

[16] IEEE Workgroup 802.1D, *IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges*, June 2004

[17] M. Schmidt, *HowTo set up common PAN scenarios with BlueZ's integrated PAN support*, http://bluez.sourceforge.net/contrib/HOWTO-PAN

[18] Standard ECMA-368 - High Rate Ultra Wideband PHY and MAC Standard 2nd edition (December 2007)

[19] Standard ECMA-369 - MAC-PHY Interface for ECMA-368 2nd edition (December 2007)

[20] Intel's UWB vision - Image from http://www.intel.com/technology/comms/uwb/

[21] RFC 3561 - "AODV - the specification" - http://tools.ietf.org/html/rfc3561

[22] RFC 3626 - "Official OLSR specification" - http://www.ietf.org/rfc/rfc3626.txt

[23] Image from R. Campos and M. Ricardo, *Dynamic and Automatic Connection of Personal Area Networks to the Global Internet*.

[24] Annex A

[25] Annex B

Annex A

PCP message definitions

In this annex are listed all the messages used by the PAN Control Protocol to create and manage the network.

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	ELECT/JOIN
Interface Type	1	Interface type: WLAN, BT or Ethernet
Local name	8	Device assigned name. ex.: "Laptop", "Desktop", "PDA"…
Interface name	8	Local interface sending this message: ex.: "eth0", "bnep1", "ath1"
BIMP	4	Numerical value used to determine my rank within the PAN

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	ACK
Interface Type	1	Interface type: WLAN, BT or Ethernet
Local name	8	Device assigned name. ex.: "Laptop", "Desktop", "PDA"…
Interface name	8	Local interface sending this message: ex.: "eth0", "bnep1", "ath1"
BIMP	4	Numerical value used to determine my rank within the PAN

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	DHCLIENT. Orders the execution of the DHCP Client

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	TREE
Number of wires	1	Indicates how many wire structures are included next in this message.
Wire 1	46	Wire structure
Wire 2	46	Wire structure
…	46	Wire structure
Wire n	46	Wire structure

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	S_TO_M. The receiver is promoted to PAN Master
Local Name	8	Device assigned name. ex.: "Laptop", "Desktop", "PDA"…
Number of wires	1	Indicates how many wire structures are included next.
Wire 1	46	Wire structure
Wire n	46	Wire structure

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	P_TO_S. The receiver is promoted to PAN Sub-Master
Local Name	8	Device assigned name. ex.: "Laptop", "Desktop", "PDA"…
Number of wires	1	Indicates how many wire structures are included next.
Wire 1	46	Wire structure
Wire n	46	Wire structure

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	S_TO_P. The receiver is demoted to PAN Peer.

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	UR_MASTER. The new arriving device is going to be Master.
BIMP	4	My BIMP (Sub Master's BIMP)
Local name	1	Device assigned name. ex.: "Laptop", "Desktop", "PDA"…
Number of wires	1	Indicates how many wire structures are included.
Wire 1	46	Wire structure
Wire n	46	Wire structure

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	UR_SUBMASTER. The arriving device is going to be Sub Master
BIMP	4	My BIMP (Master's BIMP)
Local name	1	Device assigned name. ex.: "Laptop", "Desktop", "PDA"…
Number of wires	1	Indicates how many wire structures are included next.
Wire 1	46	Wire structure
Wire n	46	Wire structure

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	UR_PEER. The new arriving device is going to be PAN Peer
Master BIMP	4	Master's BIMP
Sub-Master BIMP	4	Sub-Master's BIMP
Master name	8	Master's device name
Sub-Master name	8	Sub-Master's device name

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	NEW_DEVICE - Message sent to PAN Master from the PAN Peer which received a join message from a new arriving device.
Wire	46	Wire structure
New device BIMP	4	Arriving device's BIMP

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	ANNOUNCE
Local name	8	Device assigned name. ex.: “Laptop”, “Desktop”, “PDA”…

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	BRIDGE - Bridge setup information
Device #1 name	8	Device #1 bridge setup info is stored in the following fields
# Interfaces	1	Number of interfaces that must be bridged
Interface 1 name	8	Interface name: ex.: “eth0”, “bnep1”, “ath1” …
…	8	Interface name: ex.: “eth0”, “bnep1”, “ath1” …
Interface n name	8	Interface name: ex.: “eth0”, “bnep1”, “ath1” …
…		
Device #n name	8	Device #n bridge setup info is stored in the following fields
# Interfaces	1	Number of interfaces that must be bridged
Interface 1 name	8	Interface name: ex.: “eth0”, “bnep1”, “ath1” …
Interface n name	8	Interface name: ex.: “eth0”, “bnep1”, “ath1” …

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	ASK_BIMP

Field name	*Field size (bytes)*	*Message name / Description*
Message Type	1	ANSWER_BIMP
BIMP	4	Device’s BIMP

Annex B

Protocol Structures

In this annex are listed both *wire* and *interface* structures as they were implemented in the prototype.

```
struct wire
{
      char firstname[8];
      char secondname[8];
      char firstiface[8];
      char secondiface[8];
      unsigned char firstmac[6];
      unsigned char secondmac[6];
      unsigned char type;
      unsigned char active;
};
```

```
struct interface
{
      char name[8];
      unsigned char mac[6];
      unsigned char neighbour_mac[6];
      unsigned char type;
      unsigned char onbridge;
      int options;
};
```