

Dynamic and Automatic Interconnection of Ambient Networks supporting Mobility and Multihoming

Rui Lopes Campos
Supervisor: Manuel Pereira Ricardo

August 13, 2005

Abstract

The past and, almost, the current scenario in (mobile) communication networks is characterized by multiple networking technologies deployed independently and targetting different services, data rates, and users. Cooperation between these networks is still (almost) impossible, and when possible it does not happen in a plug and play way. On the way to next generation networks, solutions providing support for roaming between cellular and Wi-Fi networks are already available. Nevertheless, the success of future communication networks mostly depends on the full integration of these multiple networks, and new solutions need to be specified. Furthermore, in future networks new communication paradigms will come up: technology embedded in the user surrounding environment and users owning small moving networks (e.g., Personal Area Networks), instead of multiple terminals working independently, as it happens today. In this context, autoconfiguration and self-management become crucial requirements. In the past, multiple autoconfiguration mechanisms, namely for IP networks, were deployed targeting specific applications. However, in order to deal with future very dynamic scenarios the integration of these mechanism and deployment of new solutions are needed, so that dynamic autoconfiguration and self-management of devices and networks becomes possible.

The main topic of this PhD is autoconfiguration in 4G networks. We are searching for a generic autoconfiguration solution addressing the new communication environments described above; the new solution should enable dynamic and automatic interconnection of networks, and integrate existing and upcoming autoconfiguration frameworks. In spite of concentrating on the autoconfiguration issues, this PhD considers mobility, multihoming, and security as well, since they are also important features in 4G networking.

During the first year of a PhD, four major goals are expected to be met by the PhD student: a deep study of the state of the art concerning the selected research topic(s), the formulation of the problem statement, the proposal of a preliminary solution to solve the identified problem, and the work plan for the next years. In this sense, this document provides a detailed analysis of the state of the art concerning the research topics of this PhD, defines the problem statement and describes the preliminary solution, and presents the detailed work plan for the next year.

Contents

1	Introduction	5
1.1	Evolution and New Trends on (Mobile) Communication Networks	5
1.1.1	Ambient Network and Network Composition	8
1.2	The Internet as basis for the Next Generation Networks	10
1.3	Autoconfiguration in IP Networks	11
1.4	Preliminary Solution Overview	12
1.5	Structure of the Report	13
2	State of the Art	14
2.1	Autoconfiguration Frameworks in IP Networks	14
2.1.1	Dynamic Link local IPv4	15
2.1.2	IPv6 stateless Autoconfiguration	15
2.1.3	Stateless DHCP for IPv6	15
2.1.4	DHCP	16
2.1.5	PPP/IPCP	16
2.1.6	PDP Context	16
2.1.7	Autoconfiguration in MANETs	17
2.2	Multihoming and Mobility	18
2.2.1	Layer 2 Solutions	18
2.2.2	Layer 3 Solutions	19
2.2.3	Layer 3.5 Solutions	21
2.2.4	Transport Layer Solutions	23
2.2.5	Application Layer Solutions	24
2.3	Security	24
2.3.1	DHCP Authentication Option	25
2.3.2	PPP Authentication	25
2.3.3	Secure Neighbour Discovery Protocol	26
2.3.4	Extended Authentication Protocol (EAP)	26
2.3.5	Security Architecture for the Internet Protocol (IPsec)	27
2.3.6	Transport Layer Security	27
2.4	Next Steps in Signalling	28
2.4.1	NSIS Framework	28

2.4.2	GIMPS	29
3	Problem Statement	31
3.1	Autoconfiguration of Terminals	32
3.2	Autoconfiguration of Networks	33
3.3	Security	36
3.4	Mobility and Multihoming Support	36
3.5	Summary of the Problem	37
4	Preliminary Solution	38
4.1	Architectural Model	38
4.2	Basic Connectivity Manager	40
4.3	Advanced Connectivity Manager	41
4.4	GANS Transport Layer Protocol	42
4.5	Mobility, Multihoming, and Security	43
4.6	Relevant Contributions	43
5	Related Work	45
5.1	TurfNet	45
5.2	Ambient Networks Case Studies	45
5.3	MANETs autoconfiguration mechanisms	46
6	Future Work	47
7	Conclusion	49
	Bibliography	51

Acronyms

3G Third Generation

3GPP Third Generation Partnership Project

4G Fourth Generation

ACM Advanced Connectivity Manager

ACP Advanced Connectivity Protocol

ACS Ambient Control Space

AN Ambient Network

ANI Ambient Network Interface

ARI Ambient Resource Interface

ASI Ambient Service Interface

BAN Body Area Network

BCM Basic Connectivity Manager

BCP Basic Connectivity Protocol

CA Composition Agreement

C-FA Composition Functional Area

Cn-FA Connectivity Functional Area

CGA Cryptographically Generated Addresses

CSA Connectivity Service Agreement

DCCP Datagram Congestion Control Protocol

DHCP Dynamic Host Configuration Protocol

DNS Domain Name Services

DSL Digital Subscriber Line

FEUP Faculdade de Engenharia da Universidade do Porto

IETF Internet Engineering Task Force

INESC Porto Instituto de Engenharia e Sistemas de Computadores do
Porto

IP Internet Protocol

IPCP Internet Protocol Control Protocol

ISP Internet Service Provider

GANS Generic Ambient Network Signalling

GIMPS General Internet Messaging Protocol for Signaling

GGSN Gateway GPRS Service Node

GPRS General Packet Radio Service

GTLP GANS Transport Layer Protocol

GSLP GANS Signalling Layer Protocol

NAT Network Address Translator

NSIS Next Steps in Signalling

NSLP NSIS Signalling Layer Protocol

NTLP NSIS Transport Layer Protocol

PAN Personal Area Network

PhD Doctor of Philosophy

PPP Point-to-Point Protocol

QoS Quality of Service

SCTP Stream Control Transport Protocol

SEND

TCP Transport Control Protocol

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

Wi-Fi Wireless Fidelity

WLAN Wireless Local Area Network

Chapter 1

Introduction

This chapter provides an overview of the concepts and issues we are dealing with in the context of this PhD. Namely, it presents the evolution of and new trends on (mobile) communication networks, mentions the new communication paradigms that will arise in the future, demanding new network architectures, and presents the two major approaches that have been proposed for enabling next generation networks. Afterwards, it focus on the main topic of this PhD, i.e., autoconfiguration in 4G networks, describing the evolution of the Internet, why autoconfiguration became a crucial requirement as the network grew, what autoconfiguration mechanisms were deployed along the years in order to address the multiple applicability domains, and what needs to be modified and brought up to cope with the new upcoming communication paradigms. Finally, the structure of this report is presented.

1.1 Evolution and New Trends on (Mobile) Communication Networks

Future communication networks (4G networks) will be characterized by a movement towards ubiquitous communication. This includes an increasing range of wireless and wired technologies, multihomed devices, and mobility within and of networks, in addition to the mobility of the end users. In this new communication scenario, user intervention should be minimized and technology should seamlessly adapt to user's needs; communication networks should operate in plug and play way and adapt dynamically and automatically to different networks contexts, while the user is moving around. Moreover, the integration of electronic devices with computing capabilities within clothes, walls, or even in the human body will enable the creation of new computation environments and bring up new communication models, where these devices form cooperative networks, Body Area Networks (BANs), Personal Area Networks (PANs); this poses new requirements to

the (mobile) communication systems, namely in terms of autoconfiguration and self-management. Also, these devices and networks will be integrated with existing networks, such as the Internet, which have network architectures unable to deal with those new paradigms, where users will possess small moving networks, rather than multiple devices working independently and used for specific applications, e.g., mobile phone used for voice calls, laptop used for sending e-mails and/or browse the web. Thereby, new solutions must be developed to enable ubiquitous communication and render communication technologies more intelligent and adaptable to user's needs and networking contexts.

Today, (mobile) communication networks comprise multiple communication technology islands having good performance as far as their target applications are concerned, but unable to cooperate with each other automatically and dynamically, namely at the control plane. GPRS/UMTS networks [1] offer ubiquitous connectivity with relatively low data speed over a wide area. IEEE 802.11 WLANs [2] [3] [4] provide high data speed, comparable to that offered by Ethernet [5], in a small coverage area, and Bluetooth networks are well suited for creating Personal Area Network environments, besides allowing interworking with, for instance, cellular and Ethernet networks; however, manual configurations are still required. Additionally, within IP networks heterogeneity is present as well; different versions of the protocol suite operate simultaneously (IPv4, IPv6), and multiple addressing schemes (private IPv4, public IPv4, and IPv6) are defined. The key for success in future communication networks lies in the efficient combination of these and upcoming technologies.

In spite of this partitioned communication scenario, convergence between cellular and IP networks is happening; access to web pages and e-mail service through cellular networks is already possible. In addition, the current release of 3GPP [6] considers interworking between 3G and WLAN networks through cooperative authentication enabling roaming between these networks; however, functionalities such as seamless handover are not supported yet. The "all-IP" solution [7] represents a further step towards the integration of multiple existing communication technologies; it considers the migration of mobile networks towards networks based on the IP protocol, thus trying to provide standard access to any network. Yet, the focus is on data plane connectivity, and an automatic mechanism to interconnect networks in the control plane is not considered. Conversely, next generation communication systems shall integrate multiple technologies with partially overlapped coverage areas, and provide ubiquitous network service to mobile users in a plug&play way. However, providing plug and play operation and addressing the dynamics associated with future communication scenarios requires support of, for instance, autoconfiguration and self-management, multihoming, mobility, and security, in a different way than it happens today.

Along the last years, efforts have been developed in order to provide the

concrete solutions to fulfill the requirements posed by upcoming communication paradigms. These solutions can be classified in two categories: *beyond “all-IP”* and *3GPP-based*. The first considers the integration of existing networking technologies into an integrating IP-based architecture, where the different networking technologies, such as UMTS, Bluetooth, WLAN, are seen as different Layer 2 technologies that can be integrated in the overall framework by using the IP protocol; this approach is shown in Figure 1.1. Additionally, this approach defines the means for enabling interworking between the different control planes deployed for each networking technology.

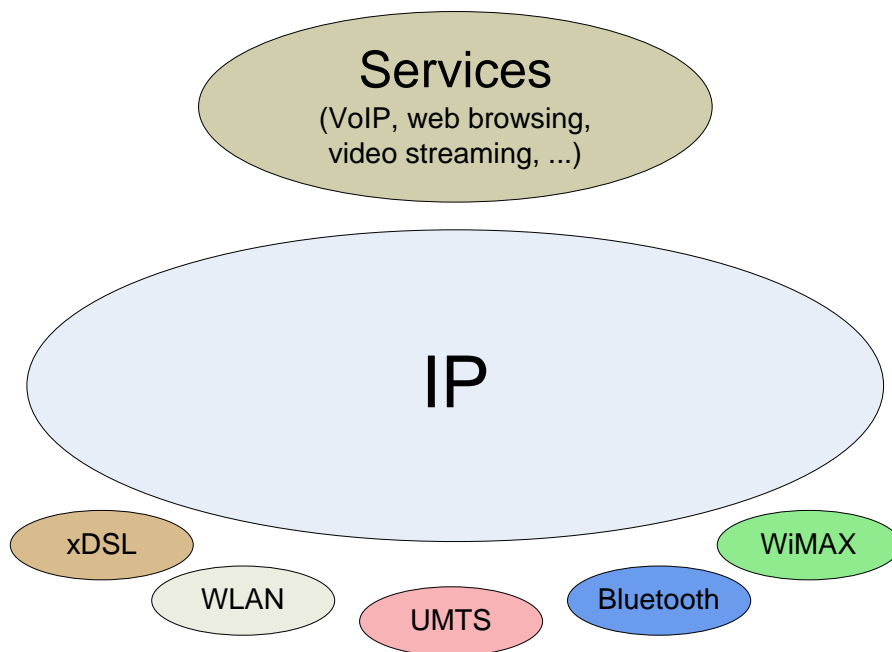


Figure 1.1: All-IP architectural model.

The second approach takes the 3GPP architecture as basis and considers access technologies other than GPRS/UMTS, e.g., WLAN, Bluetooth, as new radio access technologies that can be connected to the UMTS/3G core network, in the same way as the UMTS Terrestrial Access Network (UTRAN); this is depicted in Figure 1.2. Basically, in the first approach a new overlay architecture covering all existing networking technologies is proposed, whereas in the latter existing 3GPP architecture is reused to integrate multiple access technologies. However, in both approaches IP protocol plays a crucial role, and acts as the harmonising layer. Also, both approaches represent tentative solutions to have interworking at the control plane.

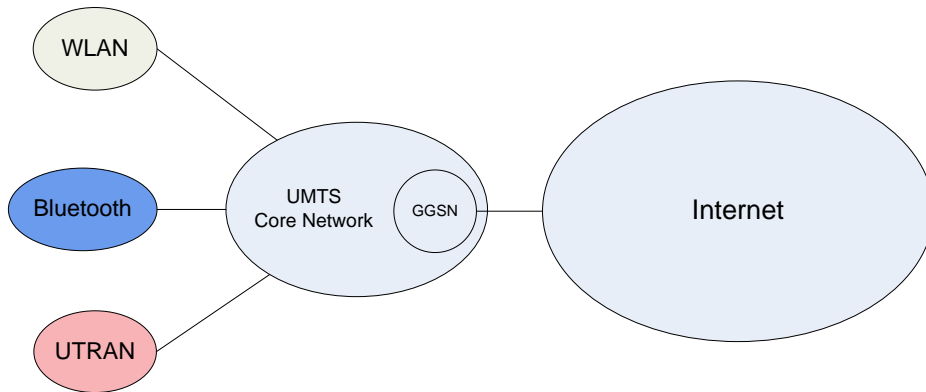


Figure 1.2: 3GPP-based architectural model.

1.1.1 Ambient Network and Network Composition

One of the solutions based on the *beyond “all-IP”* paradigm is the one being investigated in the IST Project *Ambient Networks* [8]. In this project, making part of the 6th Framework Program, solutions for plug and play co-operation between networks at both data and control plane are being studied, based on two innovative concepts, *Ambient Network (AN)* and *Network Composition*. The purpose is to allow network interworking on-the-fly and transparent to the user by using *Network Composition*. According to the AN concept, every device and network (e.g., sensors, laptops, Personal Area Networks, etc.) is treated as an AN, and the network is the primitive building block of the architecture, allowing all types of networks and devices to be composed into larger networks. An *Ambient Network* contains what is called the *Ambient Control Space (ACS)*, which comprises multiple dedicated control functions, so-called Functional Areas (FAs), dealing with specific control functions within the control plane, such as Connectivity, Mobility, Quality of Service. Furthermore, a set of three control interfaces are defined for the interaction between the *Ambient Control Space* and the outside world: the *Ambient Resource Interface*, for communicating with connectivity resources, the *Ambient Service Interface*, for interacting with services and applications, and the *Ambient Network Interface* for communicating with other networks. The latter is the interface used to perform *Network Compositions*, consisting of the negotiation of so-called Composition Agreements; this process is orchestrated by the “master” Functional Area, called Composition Functional Area (C-FA).

Network Composition allows dynamic and instantaneous interworking of networks on the control plane, in addition to the data plane cooperation possible today. Control plane interworking goes beyond basic addressing and routing by encompassing, for example, mobility, security, and QoS control. When ANs compose, they communicate across the ANI to nego-

tiate a Composition Agreement (CA), which includes the joint resources, the policies that they must follow to coordinate their ACSs, and the services provided to each other; the CA is defined by the set of FAs interested in participating in the current composition. The result of a composition can be a new composed AN that manages all logical and physical resources contributed by each constituent AN. It has its own ACS controlling all its resources, and communicates with other networks using its own identifier and via its own ANI. Composition however does not necessarily result in a new AN. In this case, resources belonging to constituent ANs stay under the control of each individual AN, and ANs just interwork based on the negotiated CA. Whether a new composed AN is created or not depends on the negotiation process, and may be influenced by different aspects, such as trust relationship between composing ANs and policies. Composition is a recursive process; it does not matter whether the composing ANs are themselves already the result of a composition.

The *Generic Ambient Network Signaling* (GANS) is the open base set of protocols enabling signaling exchange between Ambient Control Spaces via the ANI. It is important to emphasize that GANS does not replace existing or de-facto standard protocols (e.g., used to exchange autoconfiguration information). Rather, GANS is used to exchange information currently not sufficiently covered by generally accepted protocols, such as CA negotiation. The GANS protocol is currently being developed in the Ambient Networks project. Its protocol architecture is borrowed from the one already defined by the IETF NSIS WG for the NSIS protocol [9], consisting of a two-layer model: the lower layer is a general-purpose transport protocol, called GANS Transport Layer Protocol (GTLP); the upper layer comprises all signalling protocols specified for signalling exchange between pairs of Functional Areas, so-called GANS Signalling Layer Protocols (GSLPs). In addition, the GANS protocol suite is backwards compatible with the NSIS protocol suite, in the sense that every NSIS signalling application can run within the GANS Framework without modifications. NSIS is by default a protocol suite for exchanging signalling along a data path; GANS generalizes NSIS in that the control signaling is between FAs rather than along a data path.

The *Ambient Networks* project comprises all prominent European manufacturers, universities, and research institutes, as well as manufacturers and universities from outside Europe (Japan, Australia, Canada). The project is currently in its first phase, and will proceed to a second phase starting in 2006. In fact, it is one of the two most relevant IST projects in networking research in Europe, and the solutions being developed in this project are envisioned to be the enablers for 4G networking.

1.2 The Internet as basis for the Next Generation Networks

It is a consensus within the networking community, that the Internet Protocol (IP) will be the base protocol for 4G networks. In this context, the Internet will play a central role, supporting all type of multimedia services, the classical ones, such as web browsing, e-mail, video streaming, as well as the more recent and upcoming services, e.g., Voice-over-IP (VoIP), video conferencing. Basically, the Internet will become the base network to which all other networks and devices will become connected and through which they will acquire global network connectivity. However, there are well-known shortcomings related to the current Internet architecture that have to be overcome, so that it can be used as *the* underlying network over which the world-wide multimedia communication can be properly performed, and the increasing user mobility and multihomed terminals and networks can be supported.

One of these shortcomings is related to the dual-role currently associated with IP addresses, where they are used as both locators and identifier for end-hosts. In order to cope with this problem several proposals have been made during the last years, namely *FARA* [10], *HIP* [11], *PeerNet* [12], *i3* [13]. Essentially, in this proposals a new name space is defined for the identification of end-hosts and the IP address is just used as a locator; the name associated to each end-host is then dynamically bind to the specific host IP address at each moment in time, allowing implicit mobility and multihoming management. The *Layered Naming Architecture for the Internet* proposal defined in [14] goes a step further and defines a naming framework with four layers: user-level descriptors such as search keywords, e-mail addresses, etc.; service identifiers (SIDs); endpoint identifiers (EIDs), which can be compared to the new identities defined by the aforementioned solutions; and IP addresses. The authors claim three major advantages of this architecture: 1) services and data become “first-class” Internet objects, i.e., they are named independent of network location and then can freely migrate or be replicated across host and administrative boundaries; 2) mobility and multi-homing of hosts is implicitly supported; 3) network-layer and application-layer middleboxes (e.g, NATs, Firewalls) can be interposed on the data path between two communicating endpoints.

Besides the *locator-identifier* problem, other flaws are found in the current Internet architecture, which partly have to do with the early assumptions during its design three decades ago. At that time, the Internet was envisioned to be used by static users, and essentially for applications with no real-time requirements, such as file transfer applications. However, the Internet is becoming a worldwide infrastructure used for a myriad of applications, including real-time ones. Furthermore, coexistence of two IP

versions and the use of middleboxes, e.g., Firewalls and NATs, breaks the early assumed end-to-end connectivity model. Thus, new solutions have been proposed cope with these flaws. The solution proposed in [15] defines a kind of meta-control plane, called knowledge plane, for future intelligent management of the Internet. The knowledge plane has a high-level model of what the network is supposed to do, and relies on tools of Artificial Intelligence and Cognitive Systems. By augmenting the Internet with the Knowledge Plane it is expected that adaptability and reaction to changes in the required services at each time, as well as to problems, becomes faster and automatic. The major goal is to build a new generation of network, a network that can drive its own deployment and configuration. On the other hand, the solution proposed in [16] proposes the concept of *Unmanaged Internet Protocol* in order to enable self-management for the Internet, in contrast to the current scenario where the management skills from network administrators to enable end-to-end connectivity is required, e.g., NAT and Firewall configuration, IPv4-IPv6 bridging configuration. In a nutshell, this approach introduces a new name space in a similar way to *FARA* and *HIP*, for instance, and proposes a new architecture based on scalable *identity-based* routing, in contrast to the classical *locator-based* routing approach used in the current Internet architecture. By using this new architecture it is argued that the problems faced today when NATs, Firewalls, and different address domains are interposed between the source and destination endpoint can be self-managed.

1.3 Autoconfiguration in IP Networks

In the early days of the Internet, when it was just a small research network, users were networking experts capable of performing the needed network configurations. With the years, this network evolved into a huge, worldwide network, where users were no more networking experts. This implied that either users needed to acquire “know how” in networking, or that administrators needed to configure every computer on the network; this revealed to be a hard task as the networks grew. Hence, new solutions allowing automatic configuration of each computer were developed. In the early nineties the Dynamic Host Configuration Protocol (DHCP) [17] started to be deployed, providing the means for transferring configuration information to hosts on IP networks automatically. On the other hand, the worldwide spread of the Internet, made home networking a reality, and Internet Service Providers (ISPs) started to offer Internet access. First, over dial-up accesses and, more recently, over technologies such as Digital Subscriber Line (DSL). For this type of accesses, the Point-to-Point Protocol (PPP) [18] became the de-facto standard, namely for dial-up accesses, providing a method for transporting multi-protocol datagrams over point-to-point (home-to-ISP) links.

PPP defines a family of Network Control Protocols (NCPs) for establishing and configuring different network-layer protocols; the Internet Protocol Control Protocol (IPCP) [19] is defined for IP.

Foreseeing the exhaustion of IPv4 addresses, the Internet Engineering Task Force (IETF) began to work on a new version of the Internet Protocol (IPv6). In contrast to its precedent, IPv6 has built-in autoconfiguration mechanisms. The link-local address autoconfiguration provides network connectivity within a link; the global address autoconfiguration uses Router Advertisement messages [20] and provides global connectivity. The stateful autoconfiguration frameworks defined for IPv4 are also made available, namely DHCPv6 [21] and IPv6CP [22]. Furthermore, the stateless autoconfiguration framework specifies a mechanism, based on the Router Advertisement messages, through which a host learns about the proper autoconfiguration mechanism to use [23].

More recently a new communication paradigm based on ad-hoc networks came up, posing new challenges to automatic connectivity. In this new paradigm, the aforementioned stateful frameworks become difficult to implement, and alternative mechanisms have to be used. For IPv6, the built-in stateless mechanisms can be applied. Regarding IPv4, new amendments are needed, though. Thus, the solution presented in [24] came up, providing a mechanism for each terminal autoconfigure a link-local IPv4 address. Other solutions are provided in [25] [26].

In spite of these multiple IP autoconfiguration mechanisms, there is a lack of a solution which integrates them and enables the dynamic, automatic, and efficient selection of the proper mechanism according to the network context. This is actually needed in new communication environments, where users will own small moving networks composed by mobile and multihomed terminals, augmenting the number of alternatives to get connectivity, and network contexts (e.g., IP version, autoconfiguration framework available) will change rapidly due to the movement of end-users.

1.4 Preliminary Solution Overview

The preliminary solution we propose in this PhD to address the new communication models comprised in 4G networking is based on the new network architecture defined in the IST Ambient Networks project [8]. We propose an architecture for the Connectivity Functional Area (Cn-FA) comprised in the *Ambient Control Space*, in order to cope with autoconfiguration and self-management. Essentially, our solution defines a *Basic Connectivity Manager* (BCM) and an *Advanced Connectivity Manager* (ACM) which are used for establishing basic network connectivity between two ANs when they get contact with each other, and for reconfiguring a moving network (e.g., an AN) according to the network context at each time (e.g., available

points of attachment), respectively. In conjunction with these two managers, we propose a general-purpose transport layer protocol for conveying all signalling information exchanged between ACSs of different ANs, including the signalling information exchanged between ACMs running inside the ACS; BCM related signalling is not conveyed over this protocol, since it is acting earlier before IP connectivity is available; in fact, either the ACM or the general-purpose protocol need IP connectivity beforehand, and BCM is in charge of providing this. As mentioned in Section 1.1.1, the GANS protocol is defined for signalling exchange between pairs of Functional Areas running inside the *Ambient Control Space*. In our solution we specify two of these protocols, GTLP and the signalling protocol used between ACMs; the BCM is supposed to run directly over Layer 2 protocols (e.g., Bluetooth, WLAN) since it must act before IP connectivity is available, and therefore the corresponding signalling protocol messages are not transported by GTLP.

In summary, we propose a new framework to be installed within network devices and terminals that shall be able to: 1) autonomously configure the basic network connectivity when two ANs meet each other; 2) support the transport of all signalling exchanged between two ANs; 3) perform advanced connectivity management functions, such as selection of the best network interface according to user defined policies and the network context. This framework does not affect the data plane architecture, in order to allow easy migration from current devices and networks. Actually, it is assumed that the control plane functions work on behalf of the user so that plug and play and transparent connection between networks is enabled.

1.5 Structure of the Report

This report is organized in five chapters. Chapter 2 presents the state of art concerning the major topic of this PhD, network autoconfiguration, and mentions some of the solutions available to deal with mobility, multihoming, and security. Chapter 3 describes the problem statement regarding autoconfiguration in 4G networks, pointing out the need to integrate autoconfiguration with multihoming, mobility, and security solutions. Subsequently, Chapter 4 presents our preliminary solution to address the problem statement defined in the previous chapter, and Chapter 6 specifies the work plan for the next year. Finally, the conclusions are drawn in Chapter 7.

Chapter 2

State of the Art

This chapter describes the solutions in the state of the art addressing the main research topic of this PhD, network autoconfiguration. Besides, it provides a brief description of the relevant solutions defined with respect to the other topics also considered in this thesis, i.e., mobility, multihoming, and security. At the end of the chapter we present the NSIS protocol, a signalling protocol being developed in the IETF NSIS WG, which aims at providing a general-purpose protocol for signalling in the Internet.

2.1 Autoconfiguration Frameworks in IP Networks

Along the past decades, the Internet grew from a small research network, where users were networking experts, to a worldwide communication network, where all types of users, experts and non-experts, are using it. Due to the ever increasing number of non-experts users and the huge number of computers connected to a single network, manual configurations became intolerable, and an autoconfiguration solution was needed. Therefore, in an early stage, stateful autoconfiguration mechanisms were defined, namely DHCP, targetting autoconfiguration in local area networks (LANs), and PPP/IPCP, providing autoconfiguration for clients getting Internet access over dial-up connections. Later on, stateless autoconfiguration mechanisms were deployed, built-in in the new Internet Protocol version (IPv6), and most recently also defined for IPv4 for autoconfiguration link-local network connectivity. In the following, we present these autoconfigurations, dividing them up in two categories: stateless autoconfiguration and stateful autoconfiguration. Furthermore, we mention the autoconfiguration frameworks defined for a specific applicability domain, Mobile Ad-hoc Networks (MANETs).

2.1.1 Dynamic Link local IPv4

The Dynamic Autoconfiguration of IPv4 Link-Local Addresses was deployed by the IETF Zeroconf Work Group, and it is now available as an IETF RFC [24]. Multiple operating systems, such as Windows XP and Linux, already implement it as an alternative solution to DHCP. This solution defines how a host can automatically configure an IPv4 address within the 169.254/16 prefix being valid for communication with other devices in the same link. First, the host generates a random IP address in the 169.254/16 range. Next, it performs duplicate address detection using an Address Resolution Protocol (ARP) probe, in order to assess if the address is already in use; if a reply is received, it must consider that the address is being used by other terminal and try a new address. Finally, the host assigns the IP address to the local network interface, and link local connectivity becomes possible.

2.1.2 IPv6 stateless Autoconfiguration

This solution, specified in [23], defines the steps carried out by a host to autoconfigure its network interfaces in IPv6, without using a centralized service. The autoconfiguration process comprises the generation of a link-local and a global address, and a Duplicate Address Detection procedure, in order to verify the uniqueness of the autoconfigured addresses on a given link. The autoconfiguration of a link-local address is performed upon network interface activation, and after combining the well-know prefix FE80::0/10 with the interface identifier (ID), based on the MAC address; configuration of a global address is accomplished by combining the prefix announced by a local router, using the Router Advertisement messages defined in [20], with the interface ID. In addition, Router Advertisements contain two flags, M and O, informing the host whether DHCPv6 server should be contacted to acquire addresses and/or to obtain optional information.

2.1.3 Stateless DHCP for IPv6

The stateless DHCPv6 service [27] is intended to be used by nodes that have already configured an IPv6 address, through the IPv6 stateless autoconfiguration mechanism or manually, but need to acquire optional information, such as the addresses of DNS or SIP servers. This solution is a lightweight version of the stateful DHCPv6, specifying a subset of the protocol messages and avoiding state information maintenance for each individual client. It is well suited to be deployed in networking scenarios where IPv6 autoconfiguration is based on the stateless approach. To obtain the optional information, a client sends an Information-Request towards a well-known multicast address, and receives a Reply from the server containing the required information.

2.1.4 DHCP

DHCP [17] provides a framework for passing configuration information to hosts, using a client/server model. It is based on the exchange of four signalling messages. The client broadcasts a *DHCPDISCOVER* message in order to discover available servers; it may receive one or more *DHCPOFFER* messages, and after selecting one of the servers, it broadcasts a *DHCPREQUEST* containing the identification of the selected server. Finally, the server will reply with a *DHCPACK* message containing the assigned address and optional information, such as DNS and proxy server addresses [28]. With the advent of IPv6 a new DHCP version (DHCPv6) [21] came up, considering a different operation model: 1) a well-known multicast address is used by clients to address all the servers in the link, instead of broadcasts; 2) unlike DHCPv4, which is used to perform the whole host configuration, DHCPv6 can be used to just complement the stateless mechanism; 3) the messages defined for DHCPv6 are different in name and format. In real implementations, DHCPv4 and DHCPv6 are used independently for dual-stack hosts. Possible solutions to integrate the two frameworks are specified in [29].

2.1.5 PPP/IPCP

PPP is used as the standard transport framework for multiple network layer protocols over serial links. Typically it is used by a host connected to an ISP. In conjunction with other two components - a method for encapsulating multi-protocol datagrams, and a Link Control Protocol (LCP) for establishing the data-link connection - PPP defines a family of NCPs enabling auto-configuration of different network layer protocols. The IPCP [19] is defined to configure IP over PPP, namely to autoconfigure IPv4 addresses; extensions allowing configuration of optional information are specified in [30]. The protocol is based on the exchange of two messages, *Configuration-Request* and *Configuration-Ack*, and it may involve more than one negotiation round if a peer does not accept the configurations requested by the other. After IPv6 came up, the IPv6CP was defined [22]. IPv6CP specifies the same two configuration messages, and provides the means for the negotiation of an interface ID used to configure the link-local address at the local end of the link; autoconfiguration of a global address and optional information can be performed by using the stateless or stateful mechanisms above-mentioned.

2.1.6 PDP Context

Cellular networks can interwork with IP networks through a node called Gateway GPRS Support Node (GGSN). GGSN is responsible for delivering the required configuration parameters to a Mobile Station (MS) upon PDP context Activation [1]; before this, the MS shall perform a GPRS Attach

[1] to a so-called Serving GPRS Support Node (SGSN) placed between the MS and GGSN; the messages *Attach Request*, *Attach Accept*, and *Attach Complete* are exchanged. The PDP context Activation involves the three entities; the MS interacts with the SGSN which, in turn, interacts with the GGSN. The MS sends the Activate PDP Context Request, indicating the PDP type (e.g., IPv4, IPv6) and requesting an address and optional information; the SGSN contacts the GGSN in order to activate the PDP context, and the MS receives the Activate PDP Context Accept. The information carried in this message depends on the PDP type. In IPv4, it contains the assigned address and optional information, whereas in IPv6 it provides an interface ID used by the MS to configure a link-local address; the stateless or stateful mechanisms are used to configure the global address and optional information.

2.1.7 Autoconfiguration in MANETs

In the past decade, the concept of ad-hoc networking gained relevance. Ad-hoc networks based on Layer 2 technologies, such as WLAN [2] [3] [4] and Bluetooth [31], were firstly deployed enabling the creation of small networks at Layer 2 which provide the means for running applications over it, e.g., file transfer application. Nevertheless, these type of networks have some shortcomings, namely regarding establishment of IP connectivity automatically; for instance, manual configurations are still required if we want to create a PAN based on Bluetooth and run IP protocol over it. Besides Layer 2 solutions, in the last years, the concept of mobile ad-hoc networking based on the IP protocol, so-called Mobile Ad-hoc Network (MANET), came up. A MANET is defined as a group of mobile wireless nodes capable of spontaneously forming a network and support multi-hop communications thanks to the use of the IP protocol; this kind of multi-hop network allows, for instance, the extension of coverage areas or connectivity to nodes that not have suitable access technologies. This concept defines a framework for deploying networks that are able to self-manage and adapt to changes in the network context, such as network topology, thanks to the use of so-called ad-hoc routing protocols devoted to these specific environments; in [32] a survey on the current ad-hoc routing protocols, reactive and proactive, can be found. Basically, by running these protocols, each terminal belonging to a MANET is able to adapt to changes on the network topology without the manual configurations required in the earlier Layer 2 solutions. Furthermore, this new framework extends the ad-hoc networking concept to a wider scope; since it is based on the IP protocol, an ad-hoc network can be extended to multiple heterogeneous links.

MANETs are currently a hot research topic, and there are lots of research groups working on it, in order to improve the current routing protocols, provide security to these protocols, and define specific autoconfiguration solu-

tions for MANETs. In order to enable these networks to support IP services, address autoconfiguration of the ad-hoc nodes is a requirement, as we want to have the routing protocols above-mentioned working properly and minimized user involvement in the configuration process. However, currently there is no standard specification that can be used by MANET nodes to autoconfigure their IP addresses. Existing solutions for IP infrastructure-based networks, e.g., DHCP, cannot be directly applied in this context, since these protocols assume the availability of a broadcast/multicast link for signalling across a network; this type of link is not available in a MANET. Thus, several solutions for autoconfiguration in MANETs have been proposed [25] [26]. Also, a new IETF work group, called *AUTOCONF*, is being created in order to come up with a standard autoconfiguration solution for MANETs. However, current autoconfiguration mechanisms proposed for MANETs have a main shortcoming: they are normally devoted to a particular IP version, and do not provide a mechanism to cope with the heterogeneities coming up from the coexisting of the two IP versions in current IP networks.

2.2 Multihoming and Mobility

Mobility and Multihoming support considering both terminals and entire networks is a requirement for the next generation networks. Mobility and multihoming management can be performed at multiple layers and no consensus exists about the best layer to do it. In the following, we present examples of solutions providing mobility management, multihoming management, or both at different protocol layers, from Layer 2 to Application Layer.

2.2.1 Layer 2 Solutions

Some solutions managing mobility and multihoming at Layer 2, also known as Layer 2.5 solutions, since they specify a new layer over heterogeneous link-layer technologies, have been provided in the last few years. Basically, they try to manage vertical handovers (handovers between heterogeneous access technologies) and hide the multiple network interfaces that a network node may have; whether a specific terminal has one, two, or more network interfaces, it does not matter from IP layer perspective, since just a single virtual interface is presented. However, this solutions may bring up some problems, for instance, with respect to the transport layer protocols and related congestion control mechanisms, which make some assumptions that may not be fulfilled when this new approach is used. Basically, the dynamic selection of the link-layer technologies to be used at each time may quickly change the characteristics of end-to-end paths, in the extreme in a per-packet basis, which will significantly degrad performance of the transport

protocol, e.g., TCP, since associated congestion control mechanisms will have to assume the “worst” case. An example of Layer 2.5 solutions is the Generic Link Layer (GLL) [33] being developed within the IST Ambient Network project.

2.2.2 Layer 3 Solutions

Mobile IPv4/6

Mobile IPv4 (MIPv4) [34] and Mobile IPv6 (MIPv6) [35] allow a terminal (Mobile Node) to continue using its “permanent” home address as it moves around the Internet, i.e. each mobile node is always identified by its home address, independently of its current location in the Internet. To that end, two additional entities are defined: the Home Agent (HA), defined in both frameworks, and the Foreign Agent just defined for MIPv4; in MIPv6 the functionalities provided by this entity are integrated in the Mobile Node (MN). When a MN is attached to its home network everything happens as Mobility management framework did not exist. However, while away from its home network, a mobile node is also associated with a care-of address, acquired in the visited network, which provides information about its current point of attachment to the Internet. Thus, when away from home, the MN has to perform a registration with its corresponding HA in order to inform the latter about its current location. In this sense, when a Correspondent Node (CN) is willing to contact the MN, it addresses the data packets to the home address of the MN, and the HA is in charge of intercepting and forwarding these packets through a tunnel to the registred care-of address. After arriving at the end of the tunnel, each datagram is then delivered to the mobile node. The termination point of this tunnel is different for each solution. In MIPv4 it is placed at the Foreign Agent, which in turn delivers the packets to the MN, whereas in MIPv6 the tunnel is terminated at the MN; packets sent by the HA through the tunnel are de-encapsulated internally.

Hierarchical Mobile IPv6 (HMIPv6) [36] and Fast Handovers for Mobile IPv6 (FMIPv6) [37] represent optimizations of the base protocol. HMIPv6 deals with reducing the amount and latency of signalling between a Mobile Node, its Home Agent (HA) and one or more correspondents by introducing the Mobility Anchor Point (MAP) (a special node located in the network visited by the mobile node); FMIPv6 reduces packet loss by providing fast IP connectivity as soon as a new link is established by using link-layer indications to anticipate the movement of the terminal to another point of attachment.

Apart from the solutions presented in the previous paragraph, devoted to each IP version, solutions providing integration between the two are being studied, namely those referred in [38]. Furthermore, amendments to the base

solutions are proposed. For instance, the solution described in [39] proposed the joint deployment of FMIPv6 and HMIPv6 in order to benefit from all the advantages of the two schemes.

Network Mobility

Based on Mobile IP, the Network Mobility (NEMO) Basic Support protocol [40] defines a solution to support Moving Networks. The solution concerns about managing the mobility of an entire network, which changes, as a unit, its point of attachment to the Internet. The mobile network includes one or more mobile routers (MRs). The solution enables mobility support for the nodes inside the moving network and, therefore, the movement is fully transparent to the nodes inside the network. Nevertheless, in this approach all communications to and from mobile network nodes must go through the MR-HA tunnel when the mobile network is away. This results in increased length of packet route and increased packet delay. In order to overcome these shortcomings, route optimization (RO) for NEMO should be considered; an analysis on this topic is carried out in [41]. However, NEMO [40] only specifies the mechanisms to provide mobility management to IPv6 networks, and supporting only IPv6 devices and dropping IPv4 devices are not a good idea especially in the forthcoming long transition period. Thereby, in [42] an extension to the base NEMO protocol is provided in order to add IPv4 network support to the NEMO specification, and IPv4 mobile networks can operate over a base IPv6 infrastructure. NEMO solutions are envisioned to be used in existing and upcoming communication scenarios, such as Vehicular Area Networks (VANs), Personal Area Networks (PANs), Body Area Networks (BANs).

Apart from mobility support, future moving networks should be able to manage multiple network interfaces that may enable multiple communication paths to the outside, i.e., networks should be able to perform multi-homing management. In [43] the problem statement regarding this issue is defined, and a classification of the possible configurations is provided; however, no solutions are defined yet.

IP2

A different approach is provided by IP2 solution [7]. This solution defines a routing manager and a location manager located in the network control layer, allowing routing and location information management in a separate fashion. The managers are triggered from an access layer to initiate access of mobile terminals (MTs) and perform handover and location updates, between other functions. The solution defines the IP host address (IP_{ha}) to identify a Mobile Terminal, and the IP routing address (IP_{ra}) to transport packets within networks and for location information. MTs are not

aware of the IPra, as the IPHa is replaced by the IPra at the edge router (router between MT and network), and packets sent to MTs are not encapsulated. Therefore, when compared to Mobile IP protocol, the proposed routing mechanism can prevent the location information of the MT being disclosed to the peer terminal and reduce the packet header overhead caused by encapsulation used in Mobile IP.

2.2.3 Layer 3.5 Solutions

In the past few years, new solutions have been developed considering a new protocol layer between the transport layer protocols and the network layer protocols, in order to manage mobility and multihoming in a integrated manner. In the following, we present some current proposals.

Host Identity Protocol

The Host Identity Protocol (HIP) [11] defines a mechanism for decoupling the transport layer (TCP, UDP, etc) from the internetworking layer (IPv4 and IPv6), using a new Host Identity namespace. When a host uses HIP, the transport layer sockets are not bound to IP addresses but to Host Identifiers; host identifiers are public keys of a public/private key pair thus providing security implicitly. HIP provides the means for hosts to keep their communications on-going while having multiple IP addresses, either at the same time or one after another. Additionally, it allows communications to continue even when multihoming or mobility causes a change on the IP version that is available in the network, that is, if one of the communicating hosts has both IPv4 and IPv6 connectivity, either directly or through a proxy, the other host can alternate between IPv4 and IPv6, without needing to tear down and re-establish upper layer protocol connections or associations. In other words, the way upper layer protocols need to react to cross-IP-version handovers does not differ from the way they need to react to intra-IP-version handovers. In [44] it is explained how the mapping from Host Identifiers to IP addresses can be performed. This mechanism is based on a so-called rendezvous server (RVS) helping a HIP node to contact a mobile HIP node; the rendezvous server is used as the initial contact point for its clients. The clients of an RVS are HIP nodes that register their Host Identities – IP address mappings with the RVS. After this registration, other HIP nodes can get contact using the IP address of the RVS instead of the current IP address of the node they attempt to contact. Essentially, the clients of an RVS become reachable at the RVS' IP addresses. Peers can initiate a so-called HIP base exchange [11] with the IP address of the RVS, which will relay this initial communication such that the base exchange may successfully complete.

MOBIKE

Currently, the MOBIKE protocol [45] is being specified by the IETF MOBIKE (IKEv2 Mobility and Multihoming) work group (WG) and consists of the extension of the IKEv2 protocol [46], in order to enable the use of this protocol in scenarios where a specific host has multiple addresses (multihomed host), or where the IP address of the IPsec host changes (due to mobility or roaming). The current specification of the IKEv2 protocol says that IPsec and IKE Security Associations (SAs) are created between the IP address pair used during the protocol execution for establishing a SA, and after this association is created there is no way to change the addresses. Nevertheless, there are scenarios where the IP address of one of the SA endpoints may change. In this sense, the MOBIKE WG is specifying the mechanisms required to update the (outer) IP addresses associated with IKE and IPsec Security Associations (SAs), without executing the IKEv2 protocol from the scratch every the host changes its IP address due changes in the point of attachment or in the local network interface being used. By providing this extension to the base IKEv2 protocol, the MOBIKE protocol jointly supports mobility, multihoming, and security within a single framework, such as HIP.

i3

The Internet Indirect Infrastructure (i3) [13] aims at decoupling the act of sending packets from the act of receiving packets. The service model assumed is simple: source nodes send packets to a logical identifier and receivers express interest in packets sent to an identifier; in fact, this model is similar to the one used by IP multicast, where receivers express their interest in receiving packets sent to an IP multicast address associated to an IP multicast group. In the context of this solution, packets are pairs of type $(id, data)$, where the id is an m -bit identifier and data consists of a payload, usually a normal IP packet payload. Receivers express their interest in receiving packets by using triggers; typically, this triggers consist of pairs $(id, addr)$, where id is the trigger identifier, and $addr$ comprises a node's address (IP address + port number). Then, a trigger $(id, addr)$ indicates that all packets with an identifier id should be forwarded (at the IP level) by the i3 infrastructure to the node identified by $addr$. Basically, this solution proposes a new overlay network consisting of servers that store triggers and forward packets (using IP) between nodes and end-hosts. Thus, the identifier represents a logical rendezvous between the sender's packets and the receiver's trigger. This level of indirection decouples the sender from the receiver; senders need neither be aware of the number of receivers nor their location, and receivers need not be aware of the number or location of senders. Due to this properties this solution implicitly supports mobility

and multihoming management.

2.2.4 Transport Layer Solutions

In the previous sections, we have described solutions providing mobility and multihoming management, which hide changes in the address or identifier used to communicate with the peer endpoint from the transport and application protocols running above, while a host moves around or changes its active network interface. However, there might be situations wherein seamless mobility and multihoming support are just required for a subset of applications, such as VoIP and video conferencing. For that reason, solutions at the transport layer are currently available. In the following we present two examples.

Stream Control Transport Protocol

The Stream Control Transport Protocol (SCTP) [47] was originally defined to transport telephony (SS7) protocols over IP networks, with the goal of duplicating some of the reliability attributes of the SS7 signaling network in IP. This makes SCTP able to support a kind of incipient multihoming; a SCTP connection can actually have multiple IP addresses associated with it, and if a network failure occurs, the traffic being sent between connection endpoints can be seamlessly transferred to an alternate IP addresses making part of the same SCTP connection; in [48] an extension to SCTP is proposed in order to allow endpoints to use the multiple available paths for simultaneous data transmission, instead of using one at a time. Apart from multihoming support, recently extensions to the base specification has been proposed to include mobility support in the SCTP protocol as well [49] [50]. Basically, these extensions define a mechanism to dynamically update the set of IP addresses associated to a SCTP connection; this way, when a host moves towards a new IP network, it can renegotiate the set of IP addresses included in the current SCTP connection, so that the new IP address configured in the new IP network can be used for packet transmission.

Datagram Congestion Control Protocol

The Datagram Congestion Control Protocol (DCCP) [51] represents another transport layer protocol that has been extended to support mobility and multihoming. DCCP is defined as a transport protocol that provides bidirectional unicast connections of congestion-controlled unreliable datagrams; it is similar to UDP, but has built-in congestion control mechanisms. A solution proposing a preliminar design for the required extensions to the DCCP base protocol for supporting both multihoming and mobility is presented in [52]. This solution specifies a mechanism through which a moving DCCP endpoint can notify the stationary endpoint that it has moved to a

new IP address. In summary, the process is the following: the moving host sends a *DCCP-Move-Request* packet from the new IP address; the stationary host upon receiving that message sends back a *DCCP-Move-Response* to the moving host. Finally, the moving host sends a *DCCP-Move-Confirm* packet and, upon receiving this packet, the stationary node sends a *DCCP-Move-Complete* and the process completes. The description presented here is a bit simplified, since the actual process considers security mechanisms associated to the exchange of the control messages above-mentioned, in order to guarantee, for example, the authenticity of the control messages.

2.2.5 Application Layer Solutions

The application layer represents the final protocol layer wherein mobility and multihoming management may be performed. In fact, there may be some advantages in doing mobility management, at this layer, since perhaps only some applications, such as real-time applications, will actually require session continuity. In this scenario, the advantage coming up from the use of a general-purpose solution like MIPv6 is meaningless; though, we may argue that even web-browsing or file transfer applications may require mobility support, since a graceful transition between two points of attachment would allow seamless handover from the user and applications point of view. Session Initiation Protocol (SIP) [53] has been defined as the most prominent protocol to performed mobility management for multimedia applications, such as Instant Messaging applications and VoIP. For instance, in [54] it is illustrated how SIP can be used to perform mobility management. Basically, the SIP protocol considers an architecture that includes the so-called SIP Registrar and Location Services Servers, which keep track of the current location of a terminal. Therefore, a SIP client can always reach its communicating peer by using a so-called SIP Uniform Resource Indicator (URI), and network is in charge of finding the current location of the identified SIP destination.

2.3 Security

In the context of this PhD, security is analysed from two different perspectives. Firstly, we consider security associated with the autoconfiguration mechanisms described in Section 2.1; secondly, security is considered in the sense of a framework providing authentication, confidentiality, protection from Denial of Service (Dos) attacks, etc. Then, in the following, we present the particular security mechanisms defined within the context of the autoconfiguration mechanisms for IP networks; afterwards, security solutions in general are described.

2.3.1 DHCP Authentication Option

Typically, in current IP networks, when a DHCP client contacts a DHCP server to acquire configuration information, it does this without any kind of guarantee that the right DHCP server is being used; the DHCP client may be subject to denial of service attacks through the use of bogus DHCP servers, which provide the wrong parameters and preclude connectivity for the end-host running the DHCP client. On the other hand, the DHCP server does not have any guarantee about the identity of the client requesting configuration parameters; currently, authentication of the client is typically performed by confirming that the MAC address of the host requesting configuration appears in a database at the server side. Nonetheless, this method fails in authenticating the client, since the MAC address of a network interface can easily be changed and a non-authorized user can be masquerade by configuring its network interface with an authorized MAC address.

Concerning these flaws, in [55] a new option, called *Authentication Option*, to be included in each DHCPv4 message exchanged between a DHCPv4 client and a DHCPv4 server is specified. This new mechanism requires that each DHCPv4 client has a key, and it should use this key to encode any messages it sends to the server and to authenticate and verify any messages it receives from the server; the main disadvantage of this method is that the client key has to be initially distributed through some out-of-band mechanism, and should be stored locally for use in all authenticated DHCPv4 messages it exchanges with the server; in particular, this represents a disadvantage within future dynamic communication models. Furthermore, according to this solution, each DHCPv4 server must know, or be able to obtain in a secure manner, the keys for all authorized clients. With respect to DHCPv6, this mechanism is already included in the base specification [21].

2.3.2 PPP Authentication

The Link Control Protocol (LCP) defined within the PPP Framework [18] already considers an optional authentication phase, in which peer endpoints can perform mutual authentication before the actual configuration of the network layer protocol, e.g., IP, is performed. In fact, in some links it may be desirable to require a peer to authenticate itself before allowing network-layer protocol packets to be exchanged, e.g., a user connecting to Internet over a dial-up access, may have to authenticate with the ISP before it can get IP connectivity; typically this is a login/password authentication type. Within PPP authentication is not mandatory by default. If a PPP node desires that its peer authenticates with some specific authentication protocol, then it must request the use of that authentication protocol during Link Establishment phase performed by using the LCP protocol. The Extensible

Authentication Protocol (EAP) [56] is used for this. EAP is an authentication framework supporting multiple authentication methods, for instance, password based authentication; this protocol is further described in Section 2.3.4.

2.3.3 Secure Neighbour Discovery Protocol

The Secure Neighbour Discovery (SEND) Protocol [57] has been specified in order to cope with the attacks the original Neighbour Discovery Protocol (NDP) [20] is subject to. This attacks are described in [58]. For instance, a malicious node may prevent a specific host from autoconfiguring a link-local address, by sending a *Neighbour Advertisement* every time that node sends a *Neighbour Solicitation* while executing the Duplicate Address Detection (DAD) procedure [59]. Another example, may be a “rogue” router sending *Router Advertisement* messages within a link, leading to incorrect configurations and to man-in-the-middle attacks; since this router is automatically configured as the default gateway for all hosts in the current link, it will have access to all data packets sent by the hosts. The SEND protocol provides the means to overcome this attacks. Essentially, it introduces a set of new Neighbour Discovery options, in order to secure the various functions in NDP. Taking into account the two attacks aforementioned, SEND defines the following mechanisms to cope with them. Concerning the first type of attack, SEND protocol specifies the use of Cryptographically Generated Addresses (CGAs) [60] to make sure that the sender of a Neighbor Discovery message is the “owner” of the claimed address. A public-private key pair is generated by all nodes before they can claim an address. A new NDP option, the CGA option, is used to carry the public key and associated parameters. With respect to the second attack, the SEND protocol specifies the need for an entity, so-called certification path, anchored on trusted parties, to certify the authority of routers. Thus, a host must be configured with a trust anchor to which the router has a certification path before the router can be adopted as its default router.

2.3.4 Extended Authentication Protocol (EAP)

The Extensible Authentication Protocol (EAP) [56] is an authentication framework which supports multiple authentication methods. It typically runs directly over data link layers such as PPP or IEEE 802, without requiring IP. EAP may be used on dedicated links, as well as switched circuits, and wired as well as wireless links. To date, EAP has been implemented with hosts and routers that connect via switched circuits or dial-up lines using PPP. It has also been implemented with switches and access points using IEEE 802 technologies. One of the advantages of the EAP architecture is its flexibility. EAP is used to select a specific authentication mechanism, typi-

cally after the authenticator requests more information in order to determine the specific authentication method to be used. Rather than requiring the authenticator to be updated to support each new authentication method, EAP permits the use of a backend authentication server, which may implement some or all authentication methods, with the authenticator acting as a pass-through for some or all methods and peers. WLANs based on the IEEE 802.1X [61] has been a prominent applicability domain for EAP, in the last years, where EAP is used, in conjunction with other protocols, to authenticate wireless clients trying to associate to a specific WLAN access point.

2.3.5 Security Architecture for the Internet Protocol (IPsec)

The Internet architecture is by nature unsecure, and this has to do with the assumptions made during its design and early deployment. As the network evolved to a world-wide infrastructure used by millions of users, including normal and malicious users, security became a serious concern; the advent of wireless Internet raised new security problems as well. Thereby, new solutions has been proposed and standardized along the past decade to cope with these problems. The IP security protocol (IPsec) [62] is one of the solution that is standardized by the IETF. IPsec provides the means and the mechanisms to establish an secure end-to-end channel across the Internet between two IP communication peers; it is compliant with both IP versions. These framework considers three main components: an authentication header, which let's the communicating to verify whether the message was modified along the path and whether it came from the source indicated in the packet; an encapsulation security payload (ESP), that allows encryption of data against evasdropping; a negotiation protocol, called IKE (Internet key Exchange), that allows the parties to negotiate, e.g., keys, encryption algorithms, and the actual security associations (SAs). Therefore, the IPsec protocol provides the three features required to have secure communications - authentication, integrity, and confidentiality. The main advantage of this protocol is that when used it provides security to all applications in a transparent manner. One of the major applicability domains for IPsec has been in the creation of Virtual Private Networks (VPNs) over the Internet, for instance, to connect two different sites of the same enterprise, or to have an employee connecting to its enterprise network through the Internet, having the same security as if a leased line was used.

2.3.6 Transport Layer Security

The Transport Layer Security (TLS) protocol [63] provides secure communication over the Internet, which is unsecure by nature as mentioned in the previous section. The TLS protocol allows client/server applications, e.g.,

FTP, HTTP, SMTP, POP3, IMAP, to communicate in a way that is designed to prevent eavesdropping, modification of the content of exchanged messages, or message forgery. The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. At the lowest level, running over some reliable transport protocol (e.g., TCP), is the TLS Record Protocol. This protocol provides connection security that has two basic properties: the connection is private, thanks to the use of symmetric cryptography for data encryption using keys generated uniquely for each connection, and based on a secret value negotiated by another protocol (such as the TLS Handshake Protocol); the connection is reliable, since message transport includes a message integrity check using a keyed Message Authentication Code (MAC) - secure hash functions, such as SHA and MD5, are used for this. The TLS Record Protocol is used for encapsulation of various higher level protocols. Among these, is the TLS Handshake Protocol that allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys, before the application protocol (e.g., FTP, HTTP, SMTP) transmits or receives its first byte of data. One advantage of TLS is that it is application protocol independent. Higher level protocols can layer on top of the TLS Protocol transparently to the latter. The TLS protocol has been used in the Internet namely in conjunction with HTTP to form the Hypertext Transfer Protocol over Secure Socket Layer (HTTPS), which is typically used to secure World Wide Web pages for applications such as electronic commerce.

2.4 Next Steps in Signalling

2.4.1 NSIS Framework

The NSIS (Next Steps in Signalling) protocol [9] is being developed by the IETF NSIS WG which is in charge of specifying a general-purpose signalling protocol comprising multiple signalling applications; the first signalling application considered was signalling for Quality of Service (QoS), but at this time other signalling applications are also being specified, namely the NAT/Firewall signalling application used to create and delete pinholes in NAT/Firewalls for specific flows. The NSIS framework specified in [9] defines a two-layer model for this protocol: a lower layer considering a general-purpose transport protocol, called NSIS Transport Layer Protocol (NTLP), and an upper layer which may be composed by several signalling applications, called NSIS Signalling Layer Protocols (NSLPs). Apart from the two signalling application protocols referred above, the NSIS WG has already specified a solution for the transport layer, NTLP, which is called General Internet Messaging Protocol for Signaling (GIMPS); this protocol is described in next section.

2.4.2 GIMPS

The General Internet Messaging Protocol for Signaling (GIMPS) [64] is currently specified within the NSIS WG as the solution for the NSIS Transport Layer Protocol, defined in the NSIS Framework as the lower layer of the NSIS protocol suite. GIMPS represents a general-purpose transport protocol, devoted to the transportation of the signalling information it may receive from the signalling applications (NSLPs) running over it. The two main requirements identified in the NSIS Framework for the NTLP were: 1) *routing*, i.e., determine how to reach the adjacent signalling node along each direction of a data path; 2) *transport*, i.e., deliver the signalling information to that peer.

In order to address the first requirement, GIMPS defines a mechanism to discover the route towards the next GIMPS hop typically in both directions along a data path; this is performed by using a 3-way handshake (Query/Response/optional Confirm) which sets up the necessary routing state between adjacent peers. The Query message is encapsulated in a special way, depending on the message routing method (MRM) being specified by the current NSLP; the MRM defines how GIMPS should route the NSLP messages concerning a specific NSLP and specific NSLP session. For instance, if signalling along a data path is required, which is in fact the default scenario considered by GIMPS, the Query message must be sent with the IP router alert option activated. On the other hand, if signalling is supposed to be exchanged with an already known GIMPS node, the Query message can be sent directly to that node without activating the router alert option.

With respect to the second requirement, *transport*, GIMPS follows the strategy of splitting the transport problem into two layers: one has to do with the actual (secure) transportation of the signalling messages towards the right adjacent peer, and the other relates to the maintenance of all state information, and processing of the messages received from the network and from the NSLPs. Thus, GIMPS relies on well known transport protocols and security frameworks to offer reliability and security for the signalling applications messages, and specifies a specialised “messaging” layer running over these standard transport and security protocols, which is charge of managing the routing state information and discover the right adjacent peers for a NSLP message, for example. This protocol specifies two operation modes, connection mode (c-mode) and datagram mode (d-mode); TCP and UDP are currently the underlying transport protocols used for this two operation modes, respectively. Furthermore, GIMPS provides an API through which NSLPs can specify, among other things, the so-called message attributes. Three attributes are defined: reliability, security, and local processing. The first two allow an NSLP to inform GIMPS whether reliability and/or security is required. Based on these attributes GIMPS selects whether c-mode

or d-mode must be used, and whether any security framework (e.g., IPsec, TLS) needs to be considered.

The main advantage of the GIMPS protocol is its flexibility, since GIMPS is not tied to a particular message routing method, used to forward the signalling application messages; the GIMPS specification provides the guidelines to define new MRMs other than the default, more appropriate for other types of applications being defined, or which will come up in the future. Furthermore, GIMPS framework enables easy inclusion of new transport protocols and security frameworks under it without changing the overall framework, and without modifying the NSLPs running over it.

Chapter 3

Problem Statement

In the previous chapter, the state of the art on network autoconfiguration, mobility and multihoming management, and security were presented. Namely, we described the stateless and stateful autoconfiguration solutions associated to both IP versions, and showed that each mechanism has its own applicability domain. Furthermore, we exposed multiple solutions specified for supporting mobility and multihoming, and showed that this can be performed at multiple protocol layers. Finally, we presented the security mechanisms defined in conjunction with IP autoconfiguration frameworks, and described the most relevant security frameworks defined within IP networks that enable end-to-end secure communication.

Concerning the main research topic of this PhD, autoconfiguration in 4G networks, the relevant conclusion we can draw from the state of the art analysis is that the autoconfiguration mechanisms available today are not enough to deal with the very dynamic and heterogeneous scenarios envisioned for future networks, since the selection of the proper autoconfiguration framework usually requires user intervention. In addition, these new scenarios demand new solutions in order to support network self-management and deal with the different network contexts faced by devices and networks while moving around.

On the other hand, since 4G networks will be populated by multihomed devices (i.e., devices with multiple network interfaces) and mobile users, integrated support for mobility and multihoming is seen as an important requirement, so that users can seamlessly and transparently move around without noticing any change, apart from the performance changes in an environment encompassing multiple networking technologies. Besides mobility and multihoming support for devices, the same kind of support is required for entire networks, e.g., BANs, PANs, VANs.

The last, but not the less important, issue that must be considered in 4G networking is security. Security is more and more becoming a concern, specially in the Internet, which is by nature unsecure and at the same en-

visioned as the underlying transport network supporting next generation networks, as mentioned in Chapter 1. Then, the proper mechanisms providing secure communication channels and authentication of devices, networks, and users, should be integrated in the final solution for 4G networks, so that confidentiality is guaranteed and authenticity of the communicating parties is ensured.

A new solution combining the features above-mentioned is thus required. In what follows, we define the problem statement from autoconfiguration perspective considering the problems associated to autoconfiguration of terminals and autoconfiguration of networks. Then, we describe the basic requirements from mobility, multihoming, and security point of view, and argue that they should be integrated within the solution providing autoconfiguration and self-management for 4G networks.

3.1 Autoconfiguration of Terminals

The autoconfigurations performed through the mechanisms mentioned in Chapter 2 are almost static, namely for IPv4. For example, a host willing to autoconfigure its network interface will first try to contact a DHCP server and, if it does not get a reply, the configuration of IPv4 link-local address can be triggered [24]. The latter mechanism assumes that the DHCP server is not available. This raises a problem: if the server becomes available, as it can happen in very dynamic networks, such as Ambient Networks, the host will never detect the server and get global connectivity automatically. Let us consider, for instance, the PAN in Figure 3.1. Initially, IPv4 link-local addresses can be configured within the PAN. Later on, the Laptop may obtain Internet access over Ethernet and, in order to offer access to the other devices, it may support bridging between the Bluetooth and the Ethernet link. In that case, the DHCP server shown in Figure 3.1 could be used by other terminals to autoconfigure a public IPv4. However, they are not aware of the server, as no “link up” event is detected in order to run the local DHCP client, and DHCP servers do not send advertisements. In IP version 6, this problem does not exist, since by design multiple IP addresses can be assigned to an interface, and the Router Advertisements messages enable a terminal to learn about the proper autoconfiguration mechanism to use.

Considering that IPv4 and IPv6 may coexist for a long time in the Internet and that future communication scenarios will be very dynamic, the heterogeneity associated with two IP versions, the availability of multiple autoconfiguration mechanisms and its simultaneous use, implies inefficiencies and demands user intervention. A prominent future communication scenario is, for instance, a user’s Personal Digital Assistant (PDA) (dis)connecting to multiple devices (e.g., Mobile Phone, Laptop, other PDAs) and networks

(e.g., home, car, train) during a day. In this scenario, let us consider that the PDA is dual-stack. Since specific autoconfiguration solutions are deployed for each IP version, separate mechanisms for each stack and per-interface are executed, even when a single IP version is available. Focusing on a specific interface, and assuming that the PDA is connecting to a device (e.g., user's Mobile Phone) which only supports IPv6, the following process may occur. For IPv4, a *DHCPDISCOVER* is sent; after a timeout, upon no reception of a *DHCPOFFER*, the dynamic IPv4 link-local framework is run, and an ARP probe is broadcast; no ARP reply is received and the configuration is made. For IPv6, autoconfiguration of a link-local address is first accomplished; a *Neighbour Solicitation* [20] is sent to detect whether the current address is already used, and no *Neighbour Advertisement* is received. Next, a *Router Solicitation* is sent to trigger a *Router Advertisement*, which is never received. After a timeout, a *DHCP Solicit* [21] is sent. Again, no reply appears and autoconfiguration is terminated. From this analysis, we conclude that 4 messages were sent unnecessarily, as just the IPv6 link-local configuration is used. Hence, if multiple interfaces and the dynamics associated with the scenario are considered, it can be concluded that the overhead may be significant. For instance, if the terminal has four interfaces, each (dis)connecting to five different links along a period of time, an overhead of $4*5*4 = 80$ messages for a single terminal will be achieved.

3.2 Autoconfiguration of Networks

New problems come up for the autoconfiguration of networks. In order to demonstrate them we use the scenario shown in Figure 3.1. It is important to realize that some of the issues described herein are tied to the actual networking technologies being considered, namely Bluetooth. Nevertheless, the scenario allows us to expose the major shortcomings found in the state of the art, which exist, regardless the actual access technologies being used, and unable the support for the upcoming communication paradigm illustrated by this example scenario. Anyhow, in the following, we also described how things would work if another access technology (WLAN) was used inside the PAN.

Scenario Description

The network shown in Figure 3.1 represents a PAN composed of three devices: Laptop, Mobile Phone (MP) and PDA. We assume that all devices are dual-stack and support the autoconfiguration mechanisms implemented by each access network in the figure. Communication within the PAN uses Bluetooth (other access technology could be used, e.g., WLAN), and the connection to the Internet uses three technologies (UMTS, WLAN, and Ethernet). The cellular network supports IPv4 and delivers network connectivity

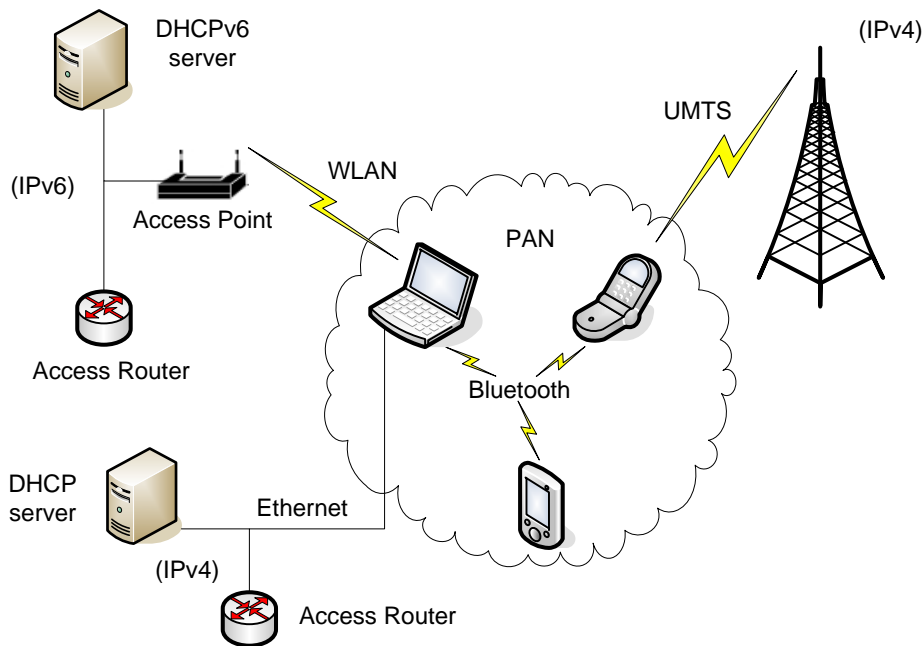


Figure 3.1: Example Scenario.

parameters to the MP upon PDP context Activation. The WLAN provides IPv6 connectivity and supports stateless IPv6 autoconfiguration along with DHCPv6 to acquire optional parameters. The Ethernet link provides IPv4 connectivity and autoconfiguration through DHCP.

Phase I – Creation of the PAN

In a first phase, the PAN is assumed to be isolated from the outside world; the dynamic configuration of IPv4 link-local addresses is carried out, in order to enable services such as file sharing; multicast DNS [65], for instance, may be used to perform local name resolution. The IP version could be pre-configured by the user or negotiated automatically between the PAN's devices using a mechanism that does not exist today. The Bluetooth PAN profile [31] is employed to create the PAN and, in particular, the Bluetooth Group Ad-hoc Network scenario is assumed; one of the terminals implements the Group Ad-hoc Network (GN) service and the others connect to it as PAN Users (PANUs); if WLAN was used the network interface cards (NICs) of all devices should simply be configured in ad-hoc mode.

Phase II – Connecting over UMTS

In a second phase, the PAN gets global connectivity over UMTS. This requires some reconfigurations. The Bluetooth PAN profile can be used, but

a different scenario has to be selected. The Bluetooth Network Access Point (NAP) scenario is selected [31]; the MP supports the NAP service and the other terminals connect to it as PANUs. Both the Laptop and the PDA have to be reconfigured in order to connect to the device offering the NAP service; if using WLAN, a new mechanism has to be defined in order to autoconfigure the Laptop and the PDA as clients of the MP with respect to the access to the UMTS network, and the Mobile Phone as the provider of this service. At the network layer further configurations are needed. From the cellular operator perspective, only the MP is known, and the requests from other terminals to autoconfigure connectivity parameters are not accepted. On the other hand, just a single IPv4 address is assigned to the MP. Then, the MP establishes a PDP context with the cellular network and acquires an address and the optional information (e.g., DNS server address). In order to grant global connectivity to the PAN, the MP is configured as a router and runs a Network Address Translator (NAT) [66]; the IPv4 link-local addresses, already configured, can still be used within the PAN. Furthermore, a *new* lightweight mechanism allowing the distribution of optional parameters, such as DNS server address, and enabling the autoconfiguration of the local routing tables of the Laptop and PDA, should be deployed.

Phase III – Connecting over WLAN

The PAN shown in Figure 3.1 is a moving network. As the PAN moves, new access networks offering better services (e.g., broader bandwidth) may be found and preferred. This is the case, when the PAN detects the WLAN access and selects it instead of the UMTS connection; reconfigurations are required again. The IP version and autoconfiguration framework supported by the new access network is different. Changes in the roles of the Bluetooth devices are needed again, namely in the Laptop, which will now act as a Bluetooth NAP; the new mechanism previous referred for autoconfiguring the three devices with respect to the new connectivity service offered by the Laptop should be used here if using WLAN. The Laptop autoconfigures itself based on the Router Advertisement messages sent by the local Access Router (AR), and collects optional parameters from the local DHCPv6 server. Concerning autoconfiguration of the PAN, the Laptop is configured as the Bluetooth NAP and as an IPv6 router between the PAN and the WLAN. Additionally, it relays the Router Advertisements received from the local AR to the PAN, so that the other devices can autoconfigure a global address. For autoconfiguration of optional information, a stateless DHCPv6 server [27] could be implemented by the Laptop; the Router Advertisements relayed by this device would inform the others about the presence of that server.

Phase IV – Connecting over Ethernet

Later on, the PAN finds out that a new access network is available. Again, reconfigurations have to happen, as IPv4 is again supported. The Laptop uses DHCP for autoconfiguring its Ethernet interface. Furthermore, the Laptop, acting as a Bluetooth NAP, needs now to operate between the Ethernet connection and the PAN, and be configured as an IPv4 router supporting NAT to offer global access to the PAN. Inside the PAN, the IPv4 configurations carried out in Phase II need to be applied, since the global connectivity is now obtained through NAT. A similar approach to that mentioned in Phase II may be employed to deliver optional information inside the PAN.

3.3 Security

Concerning security in the context of the scenario illustrated in Figure 3.1, authentication of the communication parties when they get contact to create the PAN, as well as when the PAN connects to external networks, may be required. Additionally, the need for secure channels between the multiple communicating devices may be a requirement as well, regarding both signalling and user data exchange. However, the actual problem statement with respect to security has to be further analysed, since in the first year of this PhD we concentrated more on the autoconfiguration issues. In particular, the actual security threats and attacks that may come up in this scenario will be studied in more detail after we address the problem statement related to autoconfiguration and self-management.

3.4 Mobility and Multihoming Support

Mobility and Multihoming support are in fact important requirements for next generation networks. Concerning the scenario in Figure 3.1 we can identify the need to have mobility support for the entire moving network (PAN), as well as the management of multiple connections to the outside world. In fact, if just the autoconfiguration problem is solved, the user will notice service interruption, when handing over from one point of attachment to another, and ongoing sessions will broke down. Then, considering that future communication networks shall support seamless handover between heterogeneous access technologies, and that future communication devices will possess multiple network interfaces, i.e., be multihomed, mobility and multihoming management become an important requirement in this context, and needs to be considered in every solution addressing 4G networking. However, as mentioned with respect to security, the actual problem statement and further study on how to integrate mobility and multihoming manage-

ment with the autoconfiguration mechanisms addressing 4G networking is left for further study. By now, we concentrate on the autoconfiguration problem.

3.5 Summary of the Problem

From the analysis made above, namely in sections 3.1 and 3.2, we may conclude that the multiple reconfigurations needed in the user scenario shown in Figure 3.1 cannot be performed automatically using current solutions. Furthermore, for non-expert users the deployment of this scenario is almost impossible; huge and specific configuration efforts are required. On the other hand, since the scenario is very dynamic, manual configurations are impractical; they would become invalid from time to time, leading to frequent reconfigurations. Thereby, a new framework capable of detecting changes in IP connectivity and associated autoconfiguration mechanism, and capable of configuring terminals automatically and dynamically is required. For instance, concerning the creation of the PAN in Phase I, it will be required that terminals agree upon the IP version and autoconfiguration framework to be used. Additionally, in the subsequent phases, configuration of the terminals as routers and dynamic deployment of network services (e.g., DHCP, NAT) requires knowledge of the IP version as well. Existing autoconfiguration frameworks provide configuration of IP addresses and optional information, such as DNS, SIP, or proxy servers. However, they target specific scenarios and are tied to a particular IP version. A mechanism enabling the selection of the proper IP version and autoconfiguration framework, according to the network context, and enabling network self-management is missing. If available, on the one hand, this mechanism would increase efficiency; instead of using a dual-stack and always try to autoconfigure both stacks when a new link is available, it would select the most appropriate autoconfiguration mechanism. On the other hand, it would enable dynamic and automatic interconnection between networks, such as the connection between the PAN in Figure 1 and the multiple access networks it connects to. Apart from autoconfiguration and self-management, a new solution addressing the autoconfiguration issues must consider security, multihoming, and mobility issues as well.

Chapter 4

Preliminary Solution

This chapter presents the preliminary solution we propose to solve the problems identified in the previous chapter. Our solution addresses autoconfiguration in a 4G networking environment, and relies on existing frameworks to provide mobility and multihoming support, and secure communication between communicating parties in order to grant authenticity and confidentiality; however, in an early stage, we concentrate on the autoconfiguration issues, which are the main target of this thesis, and leave for a next phase the integration of these frameworks in the overall framework. Thus, in what follows, we expose our proposal to address autoconfiguration in 4G networking. Namely, we present the architectural model of our solution and described in detail the three building blocks comprised in the proposed framework: Basic Connectivity Manager (BCM), Advanced Connectivity Manager (ACM), and GANS Transport Layer Protocol (GTLP). It is important to realize this new framework does not affect the data plane architecture, so that easy migration from current devices and networks is possible. Actually, it is assumed that this framework should work on behalf of the user so that plug and play and transparent connection between networks is enabled.

4.1 Architectural Model

In [67] we have already proposed a new framework named Meta-autoconfiguration Framework (MAF) to solve the problem described in Chapter 3, but which does not consider the *Ambient Network* and *Network Composition* concepts. Herein we propose a slightly different architecture taking these two new innovative concepts into account. We specify a concrete architecture for the Connectivity FA (Cn-FA) comprised in the ACS of an Ambient Network, by splitting MAF into a Basic Connectivity Manager (BCM) and an Advanced Connectivity Manager (ACM). Actually, the internal architecture of the Cn-FA was not yet defined in the AN project, in spite of the early efforts [68][69] performed in order to better understand the composition concept and how

the Cn-FA could operate in real user scenarios; namely, autoconfiguration and self-management issues in heterogeneous environments, where multiple address domains coexist, were overlooked; therefore, our solution goes a step further. As MAF, the Basic Connectivity Manager runs over Layer 2 protocols, whereas the Advanced Connectivity Manager runs over the GTLP defined as the framework to convey all signalling information exchanged between ACSs of different Ambient Networks. This new architecture has the advantage of benefiting from *Network Composition* concept being investigated in the AN project, since it allows interaction with the Composition Functional Area which orchestrates the Composition process. Additionally, interaction with other Functional Areas within the ACS is enabled and, thereby, integration of the solutions developed in the AN project dealing with mobility, multihoming, and security, may be easily performed in the future. Figure 4.1 presents the new architecture we propose to cope with autoconfiguration in 4G networks.

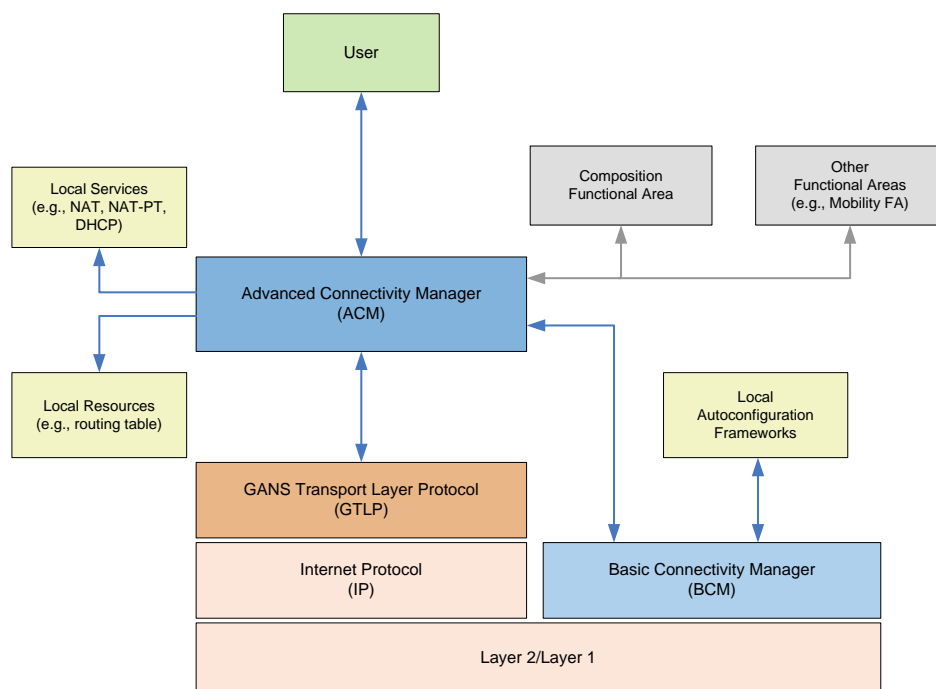


Figure 4.1: Proposed Architecture Model.

The figure above illustrates the interactions between the three building blocks of our architecture, as well as the interactions required with other local entities, such as local resources and services, and autoconfiguration frameworks (e.g., DHCP). Furthermore, in grey colour we show the interaction between the ACM and the Composition FA and the other FAs inside the ACS; however, it must be noted that in next phase of this PhD we

concentrate on the three building blocks drawn in the figure using blue and orange colour. In particular, potential integration with the “master” Composition FA will be performed in a future phase, since this FA is still just conceptually defined in the AN project.

The BCM is running over Layer 2 protocols since it must provide the basic connectivity required by the GTLP protocol and by the ACM. For that purpose, BCM interacts with the autoconfiguration frameworks locally available, and selects the mechanism negotiated a priori with the peer(s) BCM. The BCM interacts with the ACM in order to inform the latter that basic connectivity is established, pass information about the IP version and autoconfiguration mechanism used to establish the basic connectivity, and/or to start the basic connectivity establishment. The GTLP provides the transport mechanisms for the signalling information exchanged between ACMs. It is based on the General Internet Messaging Protocol for Signaling (GIMPS) proposed as a solution for the lower layer of the NSIS protocol suite [9], and extends this protocol to support abstract names and single-hop non-path related signalling, since GIMPS by default only supports multi-hop signalling along a data path [64]. The ACM runs over GTLP and uses the services offered by this protocol, which include reliability and security, to exchange signalling information with peer ACMs. Furthermore, it interacts with BCM, as mentioned above, and with local services and resources in order to realize/implement the so-called Connectivity Service Agreements (CSA) it can negotiate with peer ACMs. In fact, the negotiation of these agreements are expected to be performed within the context of a Network Composition. Nevertheless, in an early stage, we will consider that the ACMs negotiate a CSA with each other without involving the Composition FA in the process. Afterwards, we may consider the integration of this agreement in the overall Composition Agreement Framework that was roughly defined within the first phase of the AN project, and is expected to be finished in the second phase of the project. In fact, the Composition Agreement Framework defines sub-CA related to each specific Functional Area participating in it. Therefore, future integration of specific sub-CAs, such as the CSA above-mentioned becomes easy.

4.2 Basic Connectivity Manager

The BCM exists at the control plane and runs directly over Layer 2 protocols, as it must be acting before IP connectivity is available. It does not provide any autoconfiguration mechanism by itself; rather, interaction with available frameworks is expected to occur, as shown in Figure 4.1. BCM needs to interact with peer BCMs, by means of some protocol, in order to negotiate the proper IP version and autoconfiguration framework used to establish the basic connectivity, which is required by the GANS Transport Layer Protocol,

Advanced Connectivity Manager, and user applications. Concerning this protocol, called Basic Connectivity Protocol (BCP), we reuse some of the ideas we have presented in [67]. We define two messages: *ELECTION* and *NEGOTIATE*. *ELECTION* is used to elect a master device when multiple terminals are forming a network, such as in the example scenario illustrated in Figure 3.1. *NEGOTIATE* is applied in the negotiation of the IP version and autoconfiguration mechanism to be used between connecting peers.

Besides interacting with peer BCMs, the BCM has to interact with the ACM (see Section 4.3) running inside the same network device, in order to, for example, inform the ACM that a new point of attachment is available to establish connectivity, inform the ACM about the IP version and autoconfiguration framework negotiated with the peer BCMs. Let us consider the former example, in order to illustrate how these two managers may interact. Firstly, the new point of attachment would be detected by BCM upon receiving, for instance, an indication from the corresponding Layer 2 technology. Afterwards, it would notify the ACM that new point of attachment was available and this manager, in turn, would decide whether basic connectivity should be established, triggering the BCM in the positive case. Referring to the example scenario in Figure 3.1, we could explain it as follows. Concerning Phase III, the BCM running in the Laptop would detect the WLAN point of attachment and, then, it would inform the corresponding ACM that a new way to get global connectivity was available. The ACM would search its internal database in order to find out whether this new point of attachment is better than the one being used; it would find out that actually it was, and would instruct BCM to establish basic connectivity with the new external network; upon basic connectivity establishment, the BCM would notify the ACM. Subsequently, the ACM would inform its peers about the new point of attachment and, if everything went well, each BCM would be notified by the corresponding ACM about the new IP version and autoconfiguration framework that it must use to establish basic connectivity with the other nodes. Thus, interaction between the ACM and the BCM is bidirectional, and is needed to establish connectivity for the first time and to modify such connectivity due to changes in available points of attachment, which may imply changes in the IP version and autoconfiguration framework being used inside a moving network.

4.3 Advanced Connectivity Manager

Such as the BCM, the ACM also exists at the control plane. However, rather than the BCM which runs directly over Layer 2 protocols, the ACM runs over the GTLP protocol providing transport services for the signalling exchanged between ACSs belonging to different Ambient Networks. The ACM can be defined as the “master” connectivity manager which takes the decisions

based on the user preferences and needs, and based on the information it collects from the BCM running underneath; interaction between the ACM and BCM was described in the previous section. Besides interaction with the user and with the BCM, each ACM has to exchange signalling information with its potential peer ACMs; the Advanced Connectivity Protocol (ACP) is used for this purpose. Currently, the ACP protocol is neither designed nor specified; this is a goal for the next year. Interaction with peer ACMs is mostly performed in order to synchronize what are the available points of attachment to the outside for a moving network, such as the one we have illustrated in Figure 3.1. Taking into account user needs and its preferences, possibly specified by high-level policies, the ACMs agree on the best point of attachment for the entire network; this negotiation might be integrated with the composition agreement negotiation being investigated in the AN project. When the ACMs agree upon a new point of attachment for the moving network, local resources and services, e.g., routing tables, NAT, routing, bridging, may have to be configured; the ACM is in charge of doing this dynamically and automatically, taking into account the IP version and autoconfiguration framework negotiated by the BCMs.

4.4 GANS Transport Layer Protocol

Apart from the two managers, BCM and ACM, our solution comprises a general-purpose transport layer protocol, GANS Transport Layer Protocol (GTLP), for conveying all signalling information exchanged between ACSs of different ANs, including signalling exchanged between ACMs. The GANS protocol suite is being designed and specified within the Ambient Networks project for signalling exchange between pairs of Functional Areas running inside the *Ambient Control Space*. Its protocol architecture is borrowed from the one already defined by the IETF NSIS WG for the NSIS protocol [9], consisting of a two-layer model; the lower layer is defined by GTLP. GTLP is backwards compatible with the NSIS protocol suite, in the sense that every NSIS signalling application can run over it without modifications. The transport protocol currently defined by the NSIS WG, General Internet Messaging Protocol for Signaling (GIMPS) [64], assumes by default that signalling is to be performed along a data path, and requires that signalling applications use IP addresses to identify the destination endpoints. Conversely, in the AN project we have stated that signalling applications must be able to use either “symbolic” names, such as “FA-x@AN-1”, or IP addresses directly. Therefore, the GTLP takes GIMPS as basis and extends it by defining a new Message Routing Method (MRM) for GIMPS, according to the procedure specified in [64], a new mechanism for updating name binding states (mapping between symbolic names and IP addresses), as well as a new protocol, called Destination Endpoint Exploration Protocol (DEEP),

to resolve the symbolic names into the corresponding IP addresses. Then, it turns out that GTLP is composed by two sub-protocols, the so-called Extended GIMPS protocol (EGIMPS) and the DEEP protocol. The former was design and is being specified by the author in the context of the AN project, and benefited from the discussions and inputs received from other colleagues working in the team in charge of deploying the GANS protocol.

In parallel with the GTLP specification, we are currently implementing EGIMPS over NS-2 [70] in order to perform the GTLP scalability analysis during the next two months. Thus, in this context, our main contribution results from the specification of the extensions to the GIMPS protocol and the simulation of this protocol; this is mainly the result of the author's work, but benefits from discussions held in a team composed by six persons.

4.5 Mobility, Multihoming, and Security

As mentioned before, the main research topic of this PhD is autoconfiguration in 4G networking scenarios. Then, in the next year we will concentrate on this topic, and leave for the next phase (for the year after the next) the analysis of the other topics also considered in this thesis: mobility, multihoming, and security. Thereby, after we get a good solution for the autoconfiguration and self-management of a moving network, such as a PAN, we will consider these topics. Nevertheless, it must be pointed out that we will not develop new solutions addressing each of these topics. Rather we will consider the integration of existing or upcoming solutions developed by others that are able to deal with each of these issues; the multiple solutions described in Chapter 2 are candidates to be integrated with our framework.

4.6 Relevant Contributions

The relevant contributions expected at the end of this PhD are the following:

1. **Mechanism for providing dynamic and automatic connectivity between (Ambient) Networks.** This mechanism deals with the heterogeneities coming up from the two IP versions and multiple autoconfiguration frameworks coexisting in IP networks, and establishes basic network connectivity when two or more devices or networks get contact with each other. Furthermore, it is envisioned to improve the efficiency of the autoconfiguration process in a heterogeneous dynamic communication scenario where multiple (dis)connections between networks occur.
2. **Mechanism for reconfiguration of a moving network when it changes its point of attachment.** This mechanism involves execution of all the procedures needed to reconfigure a moving network,

such as a PAN, when it changes its point of attachment. Changing the point of attachment may require changes in the IP version and the corresponding autoconfiguration framework applied to autoconfigure the network interfaces of the devices belonging to the network. In addition, it may imply reconfiguration of resources (e.g., configuration as a router or a bridge) and services (e.g., NAT, NAT-PT, DHCP) in each device. This mechanism should be able to deal with these reconfigurations automatically, dynamically, and as fast as possible.

- 3. Transport Framework for interconnecting Ambient Control Spaces.** This framework will support all signalling information exchanged between pairs of Functional Areas running in different ACSs. In particular, it will support the signalling exchanged between Advanced Connectivity Managers.

Chapter 5

Related Work

Existing work related to our problem and proposal is presented below. This describes current work being developed by other people which addresses similar or the same problems we have described before.

5.1 TurfNet

TurfNet [71] represents a new network paradigm enabling the integrated operation of networks with different address realms (e.g., IPv4, IPv6), thanks to the so-called TurfNet gateways. In this context, a TurfNet is defined as an autonomous network domain, encompassing its own address space and associated control plane functions (e.g., routing, address allocation, name-to-address resolution) integrated into the TurfControl. This solution uses the emerging concept of dynamic network composition being investigated in the Ambient Networks project [8], which enables the merging of multiple TurfNets into a single one, or the simple interoperation between TurfNets. In principle, TurfNet could be used to implement, for example, the scenario given in Chapter 3. Each network in the scenario could be considered as a TurfNet comprising a TurfNet gateway to enable interoperation with other TurfNets. Nevertheless, the node implementing the TurfNet gateway inside the PAN has to change according to the different TurfNets the PAN composes with along the time, and currently the TurfNet framework does not support this feature. Rather, our solution considers dynamic and automatic selection of the gateway connecting the moving network to the outside world.

5.2 Ambient Networks Case Studies

Considering a scenario similar to ours, in [68] a solution enabling automatic and dynamic creation of a PAN and selection of the proper access network is provided. The solution is based on the concepts of Ambient Network

(AN) and Network Composition being studied in the European project Ambient Networks [8]. Nonetheless, this solution is concerned about IPv6 and does not address the heterogeneity coming up with coexistence of IPv4 and IPv6 in the current Internet. Therefore, it just solves part of the problem presented in Chapter 3.

Using a similar approach, the solution provided in [69] considers automatic and dynamic interworking between PANs by using the same two innovative concepts. However, again, a single IP version (IPv6) and particular autoconfiguration framework is considered, and the problem statement described in this report is not solved as well.

In spite of this, these two approach already define some of the ideas we also consider in our preliminary solution, namely automatic and dynamic autoconfiguration of resources and on-line negotiation of service agreement, so-called Composition Agreements. Nevertheless, in our solution we go a step further by addressing the heterogeneities coming up when the two current IP versions and the corresponding autoconfiguration mechanisms coexist in the same communication environment.

5.3 MANETs autoconfiguration mechanisms

The AUTOCONF working group has been recently created in the IETF to address autoconfiguration in Mobile Ad-hoc Networks (MANETs), which represent a particular scenario where new autoconfiguration issues come up. This group is searching for a standard autoconfiguration solution for MANETs, where each terminal is always acting as a router and running a routing protocol to maintain IP connectivity with the other terminals in the same MANET. Some autoconfiguration protocols for MANETs are already defined [25] [26], and certainly the IETF solution will be influenced by them. In contrast with our solution, MANETs require that each terminal is a router; in our approach terminals may assume different roles along the time according to the network context, and their interaction is based on the services offered and required. On the other hand, current MANETs autoconfiguration and routing protocols suffer from a problem mentioned before: separate solutions are provided concerning each IP version, and there is no mechanism enabling the selection of a proper framework considering the network context. Furthermore, there is no “intelligent” way to select the best point of attachment for a specific ad-hoc network based on user preferences and needs. In fact, the ad-hoc routing protocols define metrics associated with each discovered route, but these metrics are just based on, for instance, the number of hops or on the characteristics of the link interposed between terminals/routers; rather, in next generation network, this metrics may be defined based on policies settled by the user.

Chapter 6

Future Work

In the context of a PhD it is expected that, after the first year of his PhD, in which the state of the art is studied, the problem statement is formulated, and the preliminary solution is proposed, the PhD student starts improving the preliminary solution towards the final solution to be presented at the end of the PhD. Thus, this chapter describes the work plan for the next year of this PhD, which is centered at the preliminary solution proposed in Chapter 4.

Basically, the main goal for the second year of this PhD is to evaluate the preliminary solution we have presented herein by performing simulations. To this end, we will consider the example scenario shown in Figure 3.1 as basis for the simulations, since we actually aim at studying our solution in the context of the new communication paradigm illustrated by this scenario; anyhow, particular features of the scenario might be modified while performing the simulations.

Figure 6.1 shows a Gantt chart illustrating the work plan for the next year. Before working on the implementation for simulation, we will make a rough specification of the two managers BCM and ACM, as well as of the corresponding protocols used between BCMs and ACMs, BCP and ACP, respectively; this task will last until mid of December 2005. Afterwards, we will work on the election algorithm used by BCMs to elect a master between them; this task will be performed between mid of December 2005 and mid of January 2006. Then, we will do the implementation of the BCM and ACM over NS-2, the simulator we have selected to perform the simulations; this will last for about 4 months and is expected to be finished until the first week of April 2006. The rest of the second year will be used to perform the actual simulations. Firstly, we will perform simulations on the basic connectivity mechanism implemented by the BCM using the BCP protocol. In a nutshell, these simulations will consist of the comparison between the overhead introduced by the current autoconfiguration process, where autoconfiguration mechanisms associated to IPv4 and IPv6 run independently and at the

ID	Task Name	Start	Finish	Duration	Q4 05			Q1 06			Q2 06			Q3 06		
					Oct	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul		
1	Rough Design of BCM and ACM, and rough specification of the BCP and ACP protocols	19-09-2005	15-12-2005	64d												
2	Election algorithm for BCMs	16-12-2005	10-01-2006	18d												
3	Implementation of the BCM and ACM over NS-2	11-01-2006	07-04-2006	63d												
4	Simulation of the BCM and BCP	10-04-2006	10-05-2006	23d												
5	Simulation of the ACM and BCP without using GTLP	11-05-2006	09-06-2006	22d												
6	Simulation of the ACM and BCP using GTLP	12-06-2006	14-07-2006	25d												
7	Second year Report write-up	17-07-2006	31-07-2006	11d												

Figure 6.1: Work plan for the next year.

same time, regardless what is the IP version supported by the current point of attachment, and the overhead introduced by our solution. Secondly, we will perform the simulation on the advanced connectivity mechanism providing self-management for small moving networks, such as PANs. These simulations are actually splitted in two phases: first we will perform simulations without using the GTLP under the ACP protocol and, next, we will perform the same simulation using GTLP under ACP. The main goal here is to study the amount of time involved in the reconfiguration process when a moving network moves from one point of attachment to another; a further goal is to compare the amount of time involved in this process when GTLP is used and when it is not and the ACP runs directly over TCP or UDP. The scenario shown in Figure 3.1 will be used as the simulation scenario. Finally, after performing the simulations described above, in the second half of July 2006, we will write the second year report.

Chapter 7

Conclusion

This document reported the four main tasks developed by the author along the first year of this PhD. Namely, it presented a detail analysis of the state of the art on the select research topics, the problem statement, and the preliminary solution proposed to solve the identified problem. Basically, we identified the shortcomings found in the state of the art with respect to autoconfiguration and self-management, in order to address the new upcoming communication paradigms and cope with the heterogeneity associated to the coexistence of two IP versions and multiple autoconfiguration mechanisms in the Internet. Then, we presented a new framework to solve the identified problems, which integrates existing autoconfiguration frameworks and provides the means to reconfigure a moving network while it moves around; this framework will be further evaluated in the next year, according to the work plan presented in this document.

In conclusion, it can be said that all objectives settled in the beginning of this PhD were achieved. In the next year, we will work on the preliminary solution proposed herein by finishing its design and specification, and performing simulations and the corresponding evaluation of the results obtained. In this phase, we will concentrate on the autoconfiguration and self-management issues; integration of mobility, multihoming, and security support will be performed only afterwards, when a good solution for the autoconfiguration and self-management problem is found.

Acknowledgments

The author would like to thank the support from FCT (Fundacao para a Ciencia e a Tecnologia - Portugal) under the fellowship SFRH/BD/19429/2004/, namely the financial support for publishing an article in an international conference. Also, he would like to thank to its supervisor at FEUP and INESC Porto, professor Manuel Pereira Ricardo, for his tireless support, the valuable advices and reviews, and for his enthusiasm along the first year of this PhD.

On the other hand, the author would like to thank the support from INESC Porto for providing all the material and work conditions during this first year.

Finally, the author would like to thank to all colleagues at INESC Porto who have contributed for a good work environment and have provided some valuable comments on the work the author have been developing. Thank you all!

Bibliography

- [1] 3GPP TS 23.060. General Packet Radio Service (GPRS) Service description Stage 2, March 2005.
- [2] IEEE 802.11 Work Group. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4GHz Band. IEEE Standard, September 1999.
- [3] IEEE 802.11 Work Group. Part 11: Wireless LAN Medium Access control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band. IEEE Standard, September 1999.
- [4] IEEE 802.11 Work Group. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band. IEEE Standard, June 2003.
- [5] IEEE 802.3 Work Group. Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE Standard, 2000.
- [6] 3GPP TR 22.934. Feasibility study on 3GPP system to Wireless Local Area Network (WLAN) interworking (Release 6). 3GPP Specification, v6.2.0, September 2003.
- [7] K. Imai H. Yumiba and M. Yaburaki. IP-Based IMT Network Platform. IEEE Personal Communications, vol.8, pp. 18-23, October 2001.
- [8] Norbert Niebert et al. Ambient Networks: an Architecture for Communication Networks Beyond 3G. IEEE Wireless Communications Magazine, vol.11, pp.14-22, April 2004.
- [9] R. Hancock et al. Next Steps in Signaling (NSIS): Framework. RFC 4080, June 2005.
- [10] Aaron Falk David Clark, Robert Braden and Venkata Pingali. FARA: Reorganizing the Addressing Architecture. ACM SIGCOMM Computer Communication Review, vol.33, pp.313-321, August 2003.

- [11] R. Moskowitz et al. Host Identity Protocol. Internet Draft, draft-ietf-hip-base-03 (work in progress), June 2005.
- [12] Michalis Faloutsos Jakob Eriksson and Srikanth Krishnamurthy. Peer-Net: Pushing Peer-to-Peer Down the Stack. Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS '2003), February 2003.
- [13] Ion Stoica et al. Internet Indirection Infrastructure. Proceedings of ACM SIGCOMM '02, August 2002.
- [14] Hari Balakrishnan et al. A Layered Naming Architecture for the Internet. Proceedings of the ACM SIGCOMM Conference on Network Architectures and Protocols, Portland, OR, August 2004.
- [15] J. Christopher Ramming David D. Clark, Craig Partridge and John T. Wroclawski. A Knowledge Plane for the Internet. Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications (ACM SIGCOMM'03), pp.3-10, August 2003.
- [16] Bryan Ford. Unmanaged Internet Protocol. ACM SIGCOMM Computer Communication Review, Vol. 34, pp.93-98, January 2004.
- [17] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997.
- [18] W. Simpson. The Point-to-Point Protocol (PPP). RFC 1661, July 1994.
- [19] G. McGregor. The PPP Internet Protocol Control Protocol (IPCP). RFC 1332, May 1992.
- [20] T. Narten et al. Neighbor Discovery for IP version 6 (IPv6). Internet Draft, draft-ietf-ipv6-2461bis-03 (work in progress), May 2005.
- [21] R. Droms et al. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315, July 2003.
- [22] S. Varada et al. IP Version 6 over PPP. Internet Draft, draft-ietf-ipv6-over-ppp-v2-02 (work in progress), June 2005.
- [23] T. Narten S. Thomson and T. Jinmei. IPv6 Stateless Address Autoconfiguration. Internet Draft, draft-ietf-ipv6-rfc2462bis-07 (work in progress), December 2004.
- [24] B. Aboba S. Cheshire and E. Guttman. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927, March 2005.

- [25] C. Bernardos and M. Calderon. Survey of IP address autoconfiguration mechanisms for MANETs. Internet Draft, draft-bernardos-manet-autoconf-survey-00 (work in progress), July 2005.
- [26] Kilian Weniger and Martina Zitterbatt. Address Autoconfiguration in Mobile Ad Hoc Networks: Current Approaches and Future Directions. *IEEE Network*, vol. 18:pp. 6–11, August 2004. status:skimmed; paper-version:no; priority:5;
- [27] R. Droms. Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6. RFC 3736, April 2004.
- [28] S. Balasubramanian et al. DHCP Option for Proxy Server Configuration. Internet Draft, draft-ietf-dhc-proxysvr-opt-03 (work in progress), April 2005.
- [29] et. al T. Chown. DHCP: IPv4 and IPv6 Dual-Stack Issues. Internet Draft, draft-ietf-dhc-dual-stack-03 (work in progress), July 2005.
- [30] S. Cobb. PPP Internet Protocol Control Protocol Extensions for Name Server Addresses. RFC 1877, December 1995.
- [31] D. Snnerstam et al. Specification of the Bluetooth System (version 1.2), November 2003.
- [32] Tadeusz Wysocki Mehran Abolhasan and Eryk Dutkiewicz. A Review of Routing Protocols for Mobile Ad Hoc Networks. Elsevier, vol. 2, no. 1, pp.1-22, January 2004.
- [33] J. Sachs. A Generic Link Layer for Future Generation Wireless Networking. In Proceedings of the IEEE International Conference on Communications, 2003 (ICC '03), voll.2, pp. 834-838, May 2003.
- [34] C. Perkins Ed. IP Mobility Support for IPv4. RFC 3344, August 2002.
- [35] C. Perkins D. Johnson and J. Arkko. Mobility Support in IPv6. RFC 3775, June 2004.
- [36] et al. Hesham Soliman. Hierarchical Mobile IPv6 Mobility Management (HMIPv6). Internet Draft, draft-ietf-mipshop-hmipv6-04 (work in progress), December 2004.
- [37] Ed. R. Koodli. Fast Handovers for Mobile IPv6. RFC 4068, July 2005.
- [38] T. Larsson et al. Use of MIPv6 in IPv4 and MIPv4 in IPv6 networks. Internet Draft, draft-larsson-v6ops-mip-scenarios-01, February 2005.
- [39] et al. H. Jung. Fast Handover for Hierarchical MIPv6 (F-HMIPv6). Internet Draft, draft-jung-mobopts-fhmipv6-00 (work in progress), April 2005.

- [40] A. Petrescu V. Devarapalli, R. Wakikawa and P. Thubert. Network Mobility (NEMO) Basic Support Protocol. RFC 3963, January 2005.
- [41] C. Ng et al. Network Mobility Route Optimization Problem Statement. Internet Draft, draft-ietf-nemo-ro-problem-statement-00 (work in progress), July 2005.
- [42] K. Shima. IPv4 Mobile Network Prefix Option for NEMO Basic Support Protocol. Internet Draft, draft-shima-nemo-v4prefix-00 (work in progress), July 2005.
- [43] T. Ernst et al. Analysis of Multihoming in Network Mobility Support. Internet Draft, draft-ietf-nemo-multihoming-issues-03 (work in progress), July 2005.
- [44] L. Eggert J. Laganier. Host Identity Protocol (HIP) Rendezvous Extension. Internet Draft, draft-ietf-hip-rvs-02 (work in progress), June 2005.
- [45] P. Eronen. IKEv2 Mobility and Multihoming Protocol (MOBIKE). Internet Draft, draft-ietf-mobike-protocol-00 (work in progress), June 2005.
- [46] Charlie Kaufman. Internet Key Exchange (IKEv2) Protocol. Internet Draft, draft-ietf-ipsec-ikev2-17 (expired), September 2004.
- [47] R. Stewart et al. Stream Control Transmission Protocol. RFC 2960, October 2000.
- [48] A. Abd El Al et al. Load Sharing in Stream Control Transmission Protocol. Internet Draft, draft-ahmed-lsctp-01 (work in progress), May 2005.
- [49] M. Riegel and M. Tuexen. Mobile SCTP. Internet Draft, draft-riegel-tuexen-mobile-sctp-05 (work in progress), July 2005.
- [50] R. Stewart et al. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. Internet Draft, draft-ietf-tsvwg-addip-sctp-12 (work in progress), June 2005.
- [51] Eddie Kohler et al. Datagram Congestion Control Protocol. Internet Draft, draft-ietf-dccp-spec-11 (work in progress), March 2005.
- [52] Eddie Kohler. Datagram Congestion Control Protocol Mobility and Multihoming. Internet Draft, draft-kohler-dccp-mobility-00 (expired), July 2004.
- [53] J. Rosenberg et al. SIP: Session Initiation Protocol. RFC 3261, June 2002.

- [54] H. Schulzrinne and E. Wedlund. Application Layer Mobility using SIP. ACM Mobile Computing and Communications Review, vol.4,no.3,pp.47-57, July 2000.
- [55] R. Droms et al. Authentication for DHCP Messages. RFC 3118, March 2005.
- [56] B. Aboba et al. Extensible Authentication Protocol (EAP). RFC 3748, June 2004.
- [57] et al. J. Arkko. SEcure Neighbor Discovery (SEND). RFC 3971, March 2005.
- [58] Pekka Nikander et al. IPv6 Neighbor Discovery (ND) Trust Models and Threats. RFC 3756, May 2004.
- [59] S. Thomson et al. IPv6 Stateless Address Autoconfiguration. RFC 2462, December 1998.
- [60] T. Aura. Cryptographically Generated Addresses (CGA). RFC 3972, March 2005.
- [61] Tony Jeffree et al. IEEE Std 802.1X-2001. IEEE standard for local and metropolitan area networks - Port-based network access control, IEEE, March 2001.
- [62] S. Kent et al. Security Architecture for the Internet Protocol. RFC 2401, November 1998.
- [63] Tim Dierks and Eric Rescorla. The TLS Protocol - Version 1.1. Internet Draft, draft-ietf-tls-rfc2246-bis-13 (work in progress), June 2005.
- [64] H. Schulzrinne and R. Hancock. GIMPS: General Internet Messaging Protocol for Signaling. Internet Draft, draft-ietf-nsis-ntlp-07 (work in progress), July 2005.
- [65] S. Cheshire and M. Krochmal. Multicast DNS. Internet Draft, draft-cheshire-dnsext-multicastdns-05 (work in progress), July 2005.
- [66] P. Srisuresh and M. Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663, August 1999.
- [67] Rui Campos and Manuel Ricardo. Dynamic Autoconfiguration in 4G Networks: Problem Statement and Preliminary Solution. Accepted for publication in Proceedings of the 1st International ACM Workshop on Dynamic Interconnection of Networks (DIN'05), July 2005.

- [68] Rui Campos Cornelia Kappler, Nadeem Akhtar and Petteri Poyhonen. Network Composition using Existing and New Technologies. in Proceedings of the 14th IST Mobile & Wireless Communications Summit, Dresden, June 2005.
- [69] Rui Campos et al. Dynamic and Automatic Interworking between Personal Area Networks using Composition. Accepted for publication in the proceedings of the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'2005), March 2005.
- [70] Kevin Fall and Kannan Varadhan (Ed). The ns Manual. <http://www.isi.edu/nsnam/ns/ns-documentation.html>, August 2005.
- [71] S. Schmid et al. TurfNet: An Architecture for Dynamically Composable Networks. in Proceedings of the First IFIP TC6 WG6.6 International Workshop on Autonomic Communication, Berlin, Germany, October 2004.