Service Discovery enabling Ad Hoc networks connectivity

Jose Costa-Requena

Supervisor: Professor Raimo Kantola

Espoo, September 2004

Author:	Jose Costa-Requena	
Title of the Thesis:	Service Discovery enabling Ad Hoc networks connectivity.	
Date:	July 2004	Number of pages: 131
Faculty:	Laboratory of Networking Technology Helsinki University of Technology (HUT)	
Supervisor:	Professor: Raimo Kantola	

Ad Hoc networking is a rather old technology that is gaining a lot of momentum. However, it is still under development and there are several proposals for this technology regarding routing, addressing, connectivity and service discovery. Ad Hoc networks are self-established, the nodes need to be self-contained and have their own device discovery and control paradigms. The Ad Hoc nodes cannot rely on a fixed server informing them about the available services in the Ad Hoc network. Therefore, each node needs a discovery mechanism to learn about the network capabilities, and configure itself to the available services in the Ad Hoc network.

The goal of this thesis is to enable seamless connectivity between Ad Hoc networks and fixed IP networks (i.e. wired and wireless IP networks).

This thesis analyses the routing, the addressing and the service discovery of IP networks. Moreover, the evolution of wireless IP networks (e.g. 3G, WLAN) towards Ad Hoc networks is analyzed. A seamless connectivity and interoperability based on service discovery is proposed. The service discovery approach provides a solution for including Ad Hoc networks as an extension of the wireless IP networks. An Ad Hoc test bed, including a routing framework and a VoIP application, is implemented. This study concludes by presenting the obtained performance results.

Keywords: Service Discovery, Ad Hoc networks, Connectivity.

Preface

This thesis has been written at the Laboratory of Networking Technology at the Helsinki University of Technology in Finland. First of all, I would like to express my gratitude to the Helsinki University of Technology for giving me the opportunity to complete my licentiate studies and be on the cutting edge technology on the Telecommunications area.

I am truly thankful to my supervisor, Professor Raimo Kantola, for his interest and constant guidance, and for giving me the opportunity of doing my Master and Licentiate Thesis under his supervision.

I would like to thank all my colleagues from the Laboratory of Networking Technology, Nicklas Beijar, Kimmo Pitkaniemi, Professor Jorma Virtamo, Jorma Jormakka, Samuli Alto, Passi Lassila, Arja Hänninen, Piia, Renjish and Sampo. Special recognition to all the students I have instructed along these years and from whom I borrowed ideas and innovative perspectives (Julio Ramirez, Ignacio Gonzalez, Arto Heino, Xiaoling Zhen, Lei Xiao, Jarrod Creado, Juan Gutierrez, Javier Garcia, Ayyash Mohammad and Olmo).

From my Nokia colleagues, special thanks to Janne Lahti, Quan Young, Matti Turunen, Juha Kalliokulju, Vesa Hakkarainen, Igor Curcio and Sherkan who have strongly contributed to my career development.

Above all, I would love to thank my wife, Inmaculada Espigares, for her incessant and unconditional support given with such happiness and optimism all the time. Without her, nothing would have been the same and definitely I could not have completed this work. I would like to thank my family for staying always by my side.

August, 2004

Helsinki, Finland

Jose Costa-Requena

Table of Contents

Acronyms

2G	Second Generation cellular technology
2.5G	Represents various technology upgrades for 2G
3G	Third Generation cellular technologies
3GPP	3rd Generation Partnership Project
ABR	Area Border Router
AC	Authentication Center
AODV	Ad Hoc On Demand Distance Vector
AODV-UU	Ad Hoc On Demand Distance Vector implementation created at Uppsala University
AP	Access Point
API	Application Programming Interface
ARP	Address Resolution Protocol
AS	Autonomous System
BGP	Border Gateway Protocol
BOOTP	Bootstrap Protocol
BS	Base Station
BSC	Base Station Controller
BSS	Base Station Subsystem
BSSGP	Base Station Subsystem GPRS Protocol
BTS	Base Transceiver Station
CC	Country Code
CCRS	Common Cache and Registry Server
CCU	Channel Codec Unit
CIDR	Classless Interdomain Routing
CNAME	Canonical Name Record
CPS	Call Processing Sever

CRLF	Carriage Return followed by Line Feed
CS	Circuit Switched
CSCF	Call State Control Function
CTRIP	Circuit Switched Telephony Routing Information Protocol
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EBGP	Exterior Border Gateway Protocol
EDGE	Enhanced Data GSM Environment
EGP	Exterior Gateway Protocol
EIR	Equipment Identity Register
ENUM	E.164 numbering service
ESS	Extended Service Set
ETSI	European Telecommunications Standards Institute
FE	Forwarding Engine
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GGSN	Gateway GPRS Support Node
G-MSC	Gateway Mobile Switching Center
GMSK	Gaussian Minimum Shift Keying
GPRS	General Packet Radio Service
GSM	Global System for Mobile
GSN	GPRS Support Node
GTP	GPRS Tunneling Protocol
HLR	Home Location Register
HSCSD	High Speed Circuit Switched Data
HSS	Home Subscriber Server
HTTP	Hyper Text Transport Protocol

IBGP	Interior Border Gateway Protocol
IBSS	Independent BSS
IC	Identification Code
ICMP	Internet Control Message Protocol
I-CSCF	Interrogating Call State Control Function
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocols
IGRP	Interior Gateway Routing Protocol
IMS	IP Multimedia Subsystem
IN	Intelligent Networks
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISDN	Integrated Services Digital Network
IS-IS	Intermediate System-to-Intermediate System
ISP	Internet Service Provider
ITU	International Telecommunications Union
LBS	Location Based Services
LDAP	Lightweight Directory Access Protocol
LLC	Logical Link Control
LNP	Local Number Portability
LSA	Link State Advertisement
LSU	Link State Update
MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
MCU	Multipoint Control Unit
MED	Multi-exit Discriminator

MF	Main forwarder address
MGCF	Media Gateway Control Function
MGW	Media Gateway
MMUSIC	Multiparty Multimedia Session Control
MS	Mobile Subscriber
MSC	Mobile Switching Center
MSISDN	Mobile Subscriber ISDN
MSRN	Mobile Station Roaming Number
MX	Mail Exchange record
NAPTR	Naming Authority Pointer resource record
NDC	Destination Code
NP	Number Portability
NPDB	Number Portability Database
NSAPI	Network Layer Service Access Point Identifier
NSS	Network Switching Subsystem
OLSR	Optimised Link State Routing Protocol
OS	Operating System
OSGi	Open Services Gateway initiative
OSI	Open System Interconnect
OSPF	Open Shortest Path First
PBX	Private Branch Exchange
PCU	Packet Control unit
PDA	Personal Digital Assistant
PDP	Packet Data Protocol
PDU	Packet Data Unit
PLMN	Public Land Mobile Networks
POTS	Plain Old Telephony Service

PSTN	Public Switched Telephone Networks
QoS	Quality of Service
RFC	Request For Comments
RIP	Routing Information Protocol
RNC	Radio Network Controller
RR	Resource Record
R-SGW	Roaming Signaling Gateway
RWHOIS	Referral Whois protocol
SCN	Switched-Circuit Networks
S-CSCF	Serving Call State Control Function
SDP	Session Description Protocol
SDU	Service Data Unit
SG	Signaling Gateway
SGSN	Serving GPRS Support Node
SINR	Signal Interference noise ratio
SIP	Session Initiation Protocol
SLP	Service Location Protocol
SMTP	Simple Mail Transfer Protocol
SN	Subscriber Number
SNDCP	Sub-Network Dependent Convergence Protocol
SNMP	Simple Network Management Protocol
SOA	Start Of Authority
SOAP	Simple Object Access Protocol
SPF	Shortest Path First
SRV	Services Record
STA	Station
STD	Internet Standard

TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TID	Tunnel Identifier
TLLI	Temporary Logical Link Identifier
TRIP	Telephony Routing Information Protocol
TTL	Time To Live
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDDI	Universal Description Discovery and Integration
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication Systems
UPnP	Universal Plug and play
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VLR	Visitor Location Register
VoIP	Voice over IP
WAP GW	Wireless Application Protocol Gateway
WLAN	Wireless Local Area Network
WSDL	Web Services Description Language
WV	Wireless Village
WWW	World Wide Web
ZRP	Zone Routing Protocol

List of tables

List of Figures

Chapter 1

Introduction

Wireless communications are increasing, the bandwidth available is larger and people are getting used to having easy Internet access no matter their location. Wireless communications started with analog radio technologies (e.g. NMT or 1G networks), continued with digital radio technologies (e.g. GSM or 2G) and progressively moved towards IP based wireless networks (e.g. GPRS or 2.5G, 3G [2] and WiFi or WLAN technology [5][57]). The WiFi or WLAN radio technology defines two modes of operation: infrastructure and Ad Hoc mode.

The infrastructure mode follows the client-server approach where there is a fixed server (i.e. Access Point; AP) assigning the IP addresses to the clients, keeping synchronization of the radio channels and acting as the point of attachment to the fixed IP network. On the other hand, in Ad Hoc mode all the nodes are equal and there is no server to which the client nodes can request an IP address or gain access to the fixed IP networks. Therefore, Ad Hoc networks can be created using a specific operation mode of WLAN radio technology, where the nodes form a mesh network by themselves.

Ad Hoc networks were originally designed for military purposes but now they are re-emerging as the next generation of networks. The strength of an Ad Hoc network resides in the growth of IP over wireless and the self-organized networking feature that will enable pervasive and ubiquitous computing. Ad Hoc networks are seen as a suitable technology for embedded network devices in multiple environments such as vehicles, sensors, mobile telephones and personal

appliances. They are considered the infrastructure-less technology that will allow the users to create their Personal Area Network (PAN) anywhere and anytime. Ad Hoc networks supporting embedded networking are envisioned as a key technology for the next networking generation.

However, Ad Hoc networks present some new and unusual challenges that had not been primary concerns in fixed network deployment until now. People are used to have the power of network communications and a set of applications that rely on networking. Therefore, mobile computers and applications are becoming indispensable and requested at any time at any place, even where the appropriate infrastructures are not available. In this kind of environments, it is necessary that wireless devices learn how to communicate without routers, base stations or service providers. Wireless devices, also called mobile nodes, should perform their own network topology functions keeping track of the connection between nodes and performing routing functionality. The link state information in an Ad Hoc network changes whenever the users move, and the nodes must be able to provide automatic topology establishment and dynamic topology maintenance. Furthermore, these networks will be self-established without previous knowledge of the environment. Ad Hoc nodes need to be self-contained and have their own device discovery and control paradigms. The nodes need a set of mechanisms to allow a device to be automatically integrated and configured as part of the Ad Hoc network.

In addition to the routing problems within the Ad Hoc network, the Ad Hoc nodes cannot assume to have a fixed server that can inform about the services available in the Ad Hoc network. Therefore, each node should contain a mechanism that after joining the Ad Hoc network allows it to discover the network capabilities and configure itself to the available services. Moreover, the Ad Hoc networks have to interconnect with other IP based technologies such as wireless networks with fixed infrastructure (e.g. Wireless Local Area Networks (WLAN) and 3G networks). This interoperability requires finding the appropriate Gateway [27] or router that provides access to the fixed infrastructure (e.g. WLAN Access Points, 3G Access Nodes, etc).

Environments where multiple network technologies co-exist may create specific requirements for routing and addressing. The access mechanisms and the point of attachment from the terminal to the networks follow different procedures depending on the technology. Each wireless technology uses its own addressing space and signalling protocol. There is no full end-to-end connectivity between devices in different network technologies. Thus, when a signalling or media protocol crosses the technology border, it has to pass through a signalling, media gateway or address translation server, which converts between technologies on both sides i.e. between Internet and circuit switched networks and between private and public addresses (e.g. Network Address Translation; NAT [91]). These conversions cause delay and jitter, which degrade the routing process and the end-to-end connectivity. For that reason, it is desirable to have a mechanism to find the available services and resources in the network in order to request the appropriate services at bootstrapping (e.g. DHCP, DNS servers, etc). This approach would facilitate roaming among different network technologies, will reduce the number of conversions on the signalling and addressing layer and will enable end-to-end connectivity. Therefore, a common bootstrap service discovery mechanism for accessing multiple networks and the possibility of having a seamless IP network is desired.

The addressing in Ad Hoc and fixed IP networks follows different mechanisms. Fixed IP networks rely on a server that assigns the IP addresses, and in Ad Hoc networks the nodes have to auto assign the IP addresses. Therefore, the situation is complex when the terminals in Ad Hoc networks have their private IP address space and become part of the fixed IP network.

Moreover, after having acquired the appropriate network layer address, the application layer may use a different addressing space based on names or numbers [33], [34]. A terminal attached to an IP network may utilize the MSIDSN at application layer to initiate a call with a remote terminal. DNS servers implement the mapping between network and application layer addresses. However, in Ad Hoc networks DNS servers may not be available for doing this mapping.

Moreover, due to the mobility of the nodes a dynamic mechanism for registering the network and the corresponding application layer address into the DNS server is required. Thus, the centralized mechanism used in fixed networks cannot be applied in Ad Hoc networks.

This leads into the main problem for this thesis of defining a discovering mechanism to locate the available services in Ad Hoc networks in order to enable seamless connectivity with fixed networks. The services that would provide network connectivity would be the Gateway that connects the Ad Hoc network to the fixed infrastructure, the DHCP or NAT server, the DNS server and the SIP proxy/registrar.

Ad Hoc networks still have to meet multiple requirements in order to gain mainstream acceptance and be part of a ubiquitous IP-based network supporting traditional services such as messaging, telephony, presence and conferencing among others. Nevertheless, providing full connectivity is the main problem when considering Ad Hoc networks as part of fixed infrastructure. The connectivity problem has been solved in different ways depending on the nodes mobility. The nodes can be classified in fixed nodes (i.e. the node remains in a fixed point of attachment to the network), nomadic nodes (i.e. the node changes the point of attachment to the network but after moving to the new position it remains there for certain period of time) and mobile nodes (i.e. the node keeps changing the point of attachment to the network). This thesis discusses the different alternatives for providing connectivity in mobile environments (e.g. tunneling like in GPRS networks, Mobile IP [90], etc).

In order to get acceptance, Ad Hoc networks have to develop full connectivity with fixed infrastructure and gain access to the existing services available in fixed IP networks. This procedure should be offered to the end-user with an ease of operation and functionality. Access to different network technologies using the same terminal should be provided in a seamless manner without requiring complex settings. The terminal should provide the user with full connectivity with an ease of

operation. The terminal should be able to set up a connection with a fixed IP terminal in the public Internet or in a private Intranet, and with another wireless IP terminal with different technology such as General Packet Radio Service (GPRS), Wireless Local Area Networks (WLAN), Ad Hoc networks, etc.

In order to meet these requirements, a service discovery mechanism providing seamless connectivity between fixed and Ad Hoc networks, offering almost equivalent capabilities and network access features irrespective of the bearer or transport protocol is required. The connectivity can be provided at link layer and at application layer. The connectivity at link layer consists of obtaining a routable IP address to access the fixed infrastructure or discovering the appropriate network entity that will do the IP address translation (e.g. Network Address Translation; NAT). Obtaining a routable IP address is not an easy task since the Internet Numbering Authority (IANA) has defined the address space and indicates the address ranges used for private networks that are not routable within the public Internet. During the bootstrapping process, the mobile node can pick up its own IP address from the private address space (e.g. Class Range A in Table 1), which is not routable in the public Internet.

Table 1. IANA-allocated, non Internet-routable, IP address schemes (RFC 1918).

Address Class Range	Network Address Range
A	10.0.0.0 - 10.255.255.255
B	172.16.0.0 -172.31.255.255
C	192.168.0.0 - 192.168.255.255

Sometimes the connectivity at link layer cannot be achieved because the connection is originated from the mobile node that has a non-routable IP address. Instead the connectivity can be provided at application layer. The IP nodes can be addressed using link layer address or application layer addresses (e.g. hostnames, alias, etc). Thus when the link layer address changes, the application layer address remains the same. Therefore, the node can be globally routable using the application layer address that will be mapped into the link layer address by network components (e.g. DNS, Directory Servers, etc) or having a mobility process that

allows the node to be still reachable with the old IP address while having assigned a new address in the visiting network (e.g. Mobile IP).
The aim of this thesis is to analyse the connectivity problem and define a service discovery mechanism that will enable the connectivity at link and application layer. In concrete terms, this thesis presents an approach to implement a service discovery mechanism to provide seamless connectivity between fixed and Ad Hoc networks.

The connectivity at application layer is also analysed for the IP Telephony service. The Session Initiation Protocol (SIP) [3] is a signalling protocol that includes application layer mobility and provides a signalling infrastructure for IP Telephony, messaging, presence and conferencing services. Therefore, these services could be provided on Ad Hoc networks using SIP signaling with the in-built application layer mobility feature. SIP is specified in the Internet Engineering Task Force (IETF) Multiparty Multimedia Session Control (MMUSIC) working group [4].

Figure 1 shows different connectivity scenarios between 3G and PSTN (a), GSM (b), fixed IP (c), WLAN (d) and Ad Hoc (e) that will be analyzed in this thesis. In all the scenarios the terminal should provide similar functionality to the user. Connectivity between different networks requires in some cases to have gateways with complex functionality (e.g. A Signalling Gateway or SG in PSTN and Gateway-MSC or G-MSC in GSM networks) but in other cases the connectivity requires gateways with simpler functionality or even just IP routers with NAT functionality are needed when IP is used as the common transport in both ends of the boundary (e.g. GGSN-WLAN Access Point, which is a router with specific SIM-based authentication mechanism as required in GPRS networks, GGSN-Public IP router; ISP, etc).

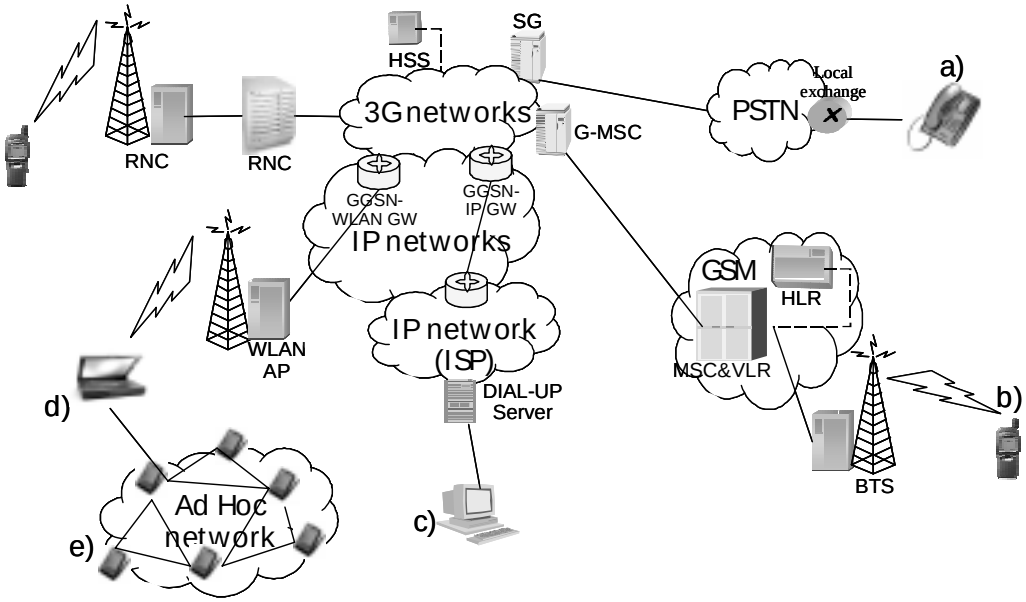

Figure 1. Connectivity between 3G networks and a) PSTN, b) GSM, c) Fixed IP, d) WLAN and e) Ad hoc networks.

To include Ad Hoc networks as part of these scenarios a service discovery mechanism to locate the services (e.g. Gateways, Mobile Routers, etc) that provide connectivity to the fixed infrastructure is required. These services may use devices with wireless cards that support multiple modes simultaneously (i.e. 802.11 in infrastructure and Ad Hoc mode), or devices with single mode (i.e. 802.11 in Ad Hoc mode) but store routing information from fixed infrastructure. This mechanism has to be integrated vertically into the lower routing layers. Thus, the service discovery becomes part of the bootstrapping mechanism and the terminal offers seamless functionality to the user across the different scenarios.

Additionally, Ad Hoc networks have to be **scalable** to support a very large number of endpoints. Ad Hoc networks need to support **network self-management** features for routing, addressing, etc. They must provide a mechanism to communicate with other Ad Hoc nodes and nodes in the fixed networks with a reasonable **Quality of Service** (QoS) requested by end points. Ad Hoc networks

need to be **extensible** to easily add new features, and support **interoperability** among different technologies.

1.1 Purpose of the Thesis

The main objective of this thesis is to present a service discovery mechanism that provides seamless connectivity between fixed and Ad Hoc IP networks. This thesis analyses the different connectivity scenarios, and the usage of service discovery as the mechanism for reduce the technology boundaries and facilitate the incorporation of new technologies such as Ad Hoc networks to become full peers to fixed IP networks.

To achieve this objective, the research documented in this thesis breaks down the problem in two well-differentiated areas such as network and application layers according to Open Systems Interconnection (OSI) networking architecture. The former area proposes a novel service discovery mechanism to locate available services in the network that enable full connectivity with the fixed infrastructure. The latter area consists of analyzing the proposed service discovery mechanism to provide connectivity at application layer using a common signaling protocol (i.e. SIP) to implement IP Telephony, Messaging and Presence services in a seamless manner between Ad Hoc and fixed IP networks.

At the network layer, the interoperability between fixed IP and Ad Hoc networks in terms of addressing, naming and routing is presented. In this layer the aim is to analyze the routing and addressing protocols, and propose a service discovery mechanism that enables connectivity on Ad Hoc IP networks towards fixed wireless networks with fixed infrastructure (e.g. GPRS, 3G, WLAN).

At the application layer the connectivity focus is on SIP functionality when it is used as the signaling protocol for providing telephony and other multimedia services in different environments. The work at the application layer concentrates on analyzing the addressing (e.g. ENUM, DNS), routing and enabling the end-to-end service deployment regardless of network access mechanism and the selected IP technology (Fixed IP or Wireless IP; GPRS, 3G, WLAN or Ad Hoc).

1.2 Structure of the Thesis

The rest of this thesis is organized as follows. Chapter 2 describes the connectivity problem and the required addressing, naming and routing protocols designed by the IETF for IP networks. This chapter presents the evolution of wireless networks towards IP based systems, and the state of the art in wireless IP technologies such as GPRS, 3G, WLAN and Ad Hoc networks is described. This chapter also presents the service discovery as it is known today implemented at application layer. Moreover, the chapter covers the addressing and connectivity with WLAN, UMTS and fixed networks.

Chapter 3 introduces the routing problems with the algorithms in Ad Hoc networks. This chapter presents a survey of existing Ad Hoc routing protocols to describe the problems in terms of routing and connectivity in Ad Hoc networks.

Chapter 4 analyses the service discovery problem in Ad Hoc networks. This part also describes the connectivity difficulties that come up when an Ad Hoc network is not isolated and instead it is attached to any of the existing wireless infrastructures (UMTS or IP fixed networks via WLAN access points). The network attachment, interoperability and service discovery in Ad Hoc networks for obtaining connectivity with the fixed infrastructure are studied. This chapter includes a new service discovery algorithm based on a novel concept for supporting the connectivity of medium and large-scale Ad Hoc networks with a fixed infrastructure.

Chapter 5 includes all the details about the design, implementation and the integration of an Ad Hoc framework to verify the service discovery mechanism under study. It solves the resource location problem within Ad Hoc networks enabling full connectivity with fixed IP networks. This Ad Hoc framework is integrated into Personal Digital Assistant (PDA) devices for testing. This implementation includes Ad Hoc routing protocols and the proposed service discovery algorithm. The performance results obtained from the tests of the routing protocols implemented in the Ad Hoc framework are also included in this chapter.

Chapter 6 presents the connectivity at session layer provided with an application layer signalling protocol (i.e. the Session Initiation Protocol; SIP) with inbuilt mobility features. This chapter includes the test on real nodes the usage of SIP protocol for implementing IP Telephony services utilising the Ad Hoc framework. The proposed service discovery mechanism to find the application layer servers (e.g. SIP servers) provides full connectivity of Ad Hoc networks to support the attachment to fixed networks and enables wireless IP telephony services regardless of the underlying transport for the wired or wireless IP network. This chapter describes the structure of the SIP messages (requests and responses) and includes examples to clarify the protocol functionality. The main benefits of the SIP protocol are modularity, transport protocol neutrality, mobility, extensibility and the possibility to create new services. Some extensions added to SIP in order to deploy new services like location and presence information are presented. Performance results of IP Telephony tests using the Ad Hoc framework are also presented in this chapter.

Chapter 7 presents our conclusions and a summary of the work.

Chapter 2

IP networks connectivity

This chapter describes the connectivity model provided in IP networks. This model is based on addressing and routing protocols that apply for fixed and wireless IP networks. This chapter also covers the IP networks built on top of wireless technologies. The General Packet Radio (GPRS) networks or 2.5G wireless networks that already provide IP connectivity. Further on, UMTS or 3G networks provide a fully transparent access to Internet and their own infrastructure is built on top of IP networks. We analyse the new wireless IP technologies such as Wireless Local Area Networks (WLAN) and Ad Hoc networks that also require interconnecting with the existing wireless infrastructures. Actual WLAN and Ad Hoc networks in the market are mostly based on the "802.11" radio technology [5] but Ad Hoc networks can be implemented on other radio bearers such as Blueetooth. The nodes that compose the Ad Hoc network can be randomly distributed and they communicate with each other without having any specific role. This chapter describes the addressing problems created when Ad Hoc wireless networks are connected to IP wireless networks with fixed infrastructure.

2.1 IP connectivity model

The Internet architecture was designed using a simple but robust technology for setting up a wide area network, for communicating among multiple and diverse endpoints. The basic Internet philosophy consists of a modular design, composed of independent but interconnected protocols (ARP, IP, RIP, OSPF, etc) that

provide an overall networking technology. The simplicity of this architecture grounds on the fact that the IP protocol hides the underlying bearer. The IP protocol provides a mechanism for deploying robust networks independently of the physical layer. IP creates best effort networks where the strength resides in the diversity of computer networking and the growth of devices that IP patches together. The IP based communication is increasing, the bandwidth available is larger and people are getting used to have easy access no matter of their location. This has been the main reason of the momentum behind the IP networks. Nevertheless, the lack of coherence in the growing Internet may convert the success of the IP networks into a chaotic technology. The new Internet design is moving "*from a world of a single, coherent architecture designed by a small group of people, to a world of a complex, intricate architecture to address a wide-spread and heterogeneous environment*" [6]. The Internet is growing by aggregating individual pieces of the architecture designed by sub-communities without paying attention to how each piece fits into the larger picture.

The Internet architecture is based on a modular concept where even the basic addressing and routing mechanisms are based on multiple protocols that put together provide the overall functioning. This modular concept is the design guideline behind the Internet architecture that allows lots of new networks (wireless, fixed) adopting IP as the protocol. This conforms to the idea that "heterogeneity is inevitable and must be supported by design" [7] on new protocols. This means that a new design should be able to handle multiple and diverse environments and devices. Therefore, when designing a new protocol for continuing the assistance of the core IP protocols in their functioning, *Complexity, Overhead, Resilience, Performance* and *Security* are the main issues to consider. These are the guidelines extracted from the Internet design that we have kept in mind when designing the new protocols proposed in this thesis.

2.2 Network and Session layer connectivity

The Internet was designed with the aim of implementing a robust infrastructure for network communications. The argument for the Internet was to provide an end-to-

end design based on distributed systems, where the end points have the final decision. Networking functions should be delegated as much as possible outside the network. The second premise of the Internet architecture is that the "internet-work" is built by layering a unique "internet-working protocol" on top of various network technologies in order to interconnect various and different networks. Thus the Internet architecture is designed using the concept of building blocks to create a robust and reliable infrastructure. This concept yields to the idea that the routing responsibility is set on top of different protocols that can evolve independently according to their needs. The following sections differentiate the two levels of connectivity provided in IP networks; network and session layer connectivity. Figure 2 presents the location of the protocols involved in the IP model according to the OSI layer model.

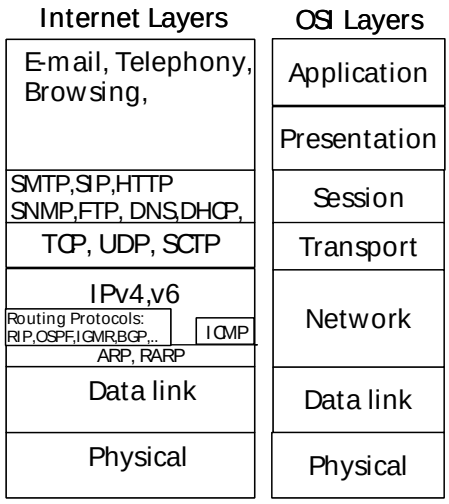

Figure 2. The layering of different protocols within an IP node.

2.2.1 Network layer connectivity

In IP networks, the network layer provides connectivity between any pair of hosts in a connected network. Figure 3 shows the network layer connectivity, where the intra-domain routing protocols, such as RIP [8] and OSPF [9], create and maintain routing tables for hosts in a single Autonomous System (AS).

Figure 3 also depicts the inter-domain routing protocols of which BGP-4 [10] is used today to distribute routes between Autonomous Systems. Every host is able to

send signalling messages and establish connections to all other hosts in the network that are identified by IP addresses. The IP addresses are network layer addresses that are either 32-bit or 128-bit binary strings, depending on whether IPv4 or IPv6 is used. The connectivity is achieved at network layer by exchanging the IP address among routers.

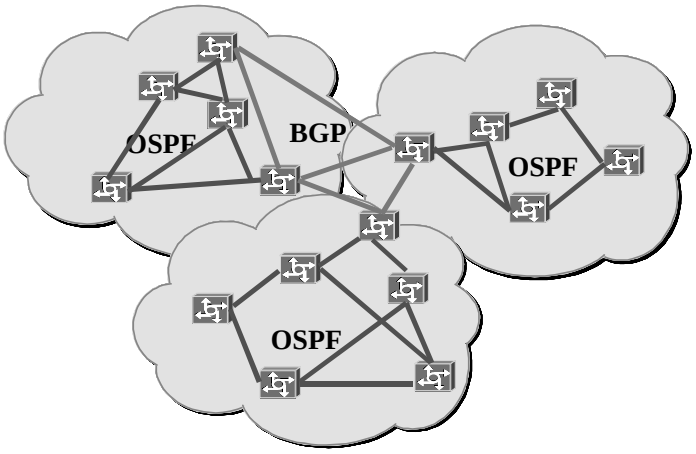

Figure 3. Network architecture with BGP/Inter and OSPF/Intra-networks routing protocols.

2.2.2 Session layer connectivity

The IP addresses create usability barriers to identify the terminating device. It is much more user friendly to use *textual names* and for this reason the Domain Name System (DNS) [11], [12] was designed to overcome this usability problem. DNS maps textual host names into IP-addresses (IPv4 or IPv6) and it is easier to identify hosts or users with a text string. DNS provides addresses resolution for any service that requires this kind of conversion such as “e-mail” and “browsing”.

Therefore, the session layer connectivity is achieved by having a higher level naming space that is available for applications servers (e.g. mail server, browsers, SIP servers, etc). The application layer naming space is mapped into IP address using DNS or other directory based services (e.g. LDAP). This allows the nodes can change the IP address while keeping the same name. Figure 4 shows the DNS infrastructure based on hierarchy of distributed DNS servers that allow the mapping of names into IP addresses supporting session level connectivity. Figure 4

depicts a couple of examples where the e-mail addresses or web addresses are mapped first into IP addresses (e.g. transaction 1) in the figure where the node contacts the nearest DNS server for obtaining the IP address of the e-mail or web server) and after that the transaction can take place (i.e. transaction 2) in the figure where the node does an e-mail transaction with the remote mail server or download a web page from the web server).

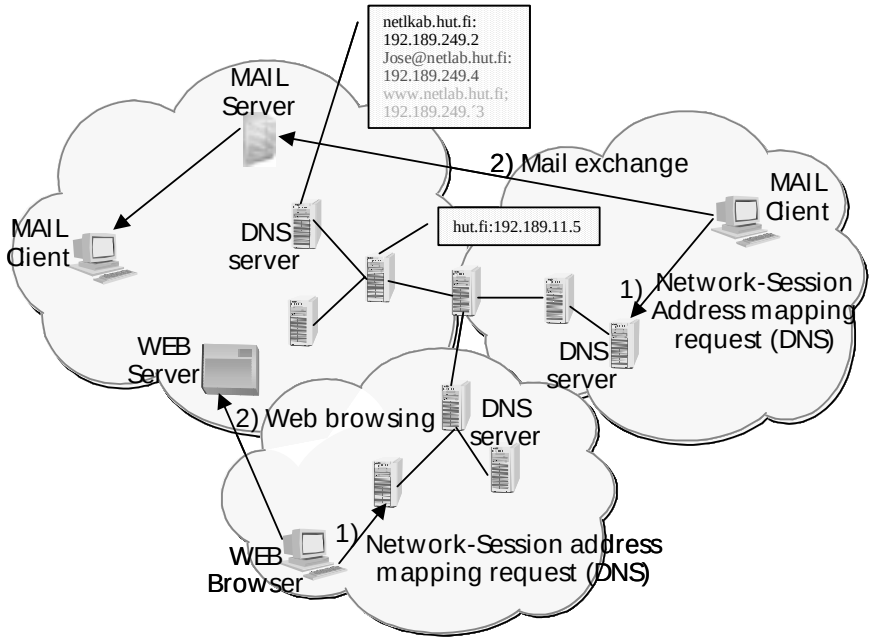

Figure 4. Network architecture with DNS hierarchy and interaction name-IP mapping.

The Internet also suffers from addressing problems. The lack of available IP addresses for the increasing number of nodes forced the assignment of IP addresses blocks for private and public usage. The public IP addresses are routable within the public domain while the private addresses can only be used within closed networks. A network element named Network Address Translator (NAT) was defined to do automatic conversion of private into public addresses to enable the connectivity of closed networks with the public Internet. This is an important issue that limits the scalability of IP networks. IP protocol version 6 (IPv6) defines a wider range of addresses, however in IPv4 NAT is already a legacy component to deal with (The NAT functionality is described in Section 2.1.2.6). Moreover, Ad Hoc networks suffer from this addressing problem because the nodes themselves assign their IP addresses from the private IP address range. Thus, the

interoperability of Ad Hoc nodes with public Internet domain requires NAT functionality, or a service discovery that locates the node that provides public domain addresses (e.g. DHCP, WLAN Access Point, etc), or the gateway towards the public Internet. Therefore, the Ad Hoc networks can be attached to the wireless or fixed infrastructure behind a NAT and the only connectivity that could be provided would be at session layer. In order to have session layer scenarios we have to obtain network layer connectivity first. Therefore, the rest of this chapter focus on analysing the connectivity at network layer while the session connectivity would be analysed in chapter 6 where SIP is used as the application for providing connectivity.

2.3 Network layer connectivity scenarios

The following sections describe the addressing and routing protocols defined in the IETF to provide connectivity at network layer to understand Ad Hoc networks.

The first step for obtaining the network layer connectivity consists of obtaining an IP address. The IP address can be hard coded in the device or it can be obtained dynamically from a *Dynamic Host Configuration Protocol* (DHCP) [19] server.

2.1.2.1 Dynamic Host Configuration Protocol

The *Dynamic Host Configuration Protocol* (DHCP) [19] is an Internet protocol to automatically assign IP addresses. Moreover, DHCP delivers other TCP/IP stack configuration parameters such as the subnet mask, default router, and other configuration information such as printer addresses, time and news servers, SIP servers, etc.

BOOTP [59] protocol is the predecessor of DHCP for automatically assigning configuration information into the host during the bootstrapping process, normally for hosts without hard disk for storing all this configuration information. DHCP is based on the client-server model, where the DHCP server stores network addresses and delivers configuration parameters to dynamically configured hosts.

DHCP implements the “lease” concept according to which addresses are allocated dynamically to clients that will be temporarily connected to the network. This allows sharing a limited pool of IP addresses among a group of clients that do not need permanent IP addresses. BOOTP protocol was assigning fixed addresses to specific devices when they where booting because at the time when BOOTP was conceived the IP address space was bigger than the demand from IP clients.

DHCP incorporates a dialogue during the IP address assignment, which provides other options during the process. BOOTP protocol only allowed two types of messages (request and reply). DHCP has seven possible message types that can be used during the address assignment procedure.

To obtain an address

The host that incorporates a DHCP client connects to the network and broadcasts a DHCPDISCOVER packet. The DHCP servers on the local segment receive the broadcast and return a DHCPOFFER packet that contains an IP address and additional information. The servers may initiate a preliminary checking such as generating an ARP or ICMP echo to ensure that the address they are going to assign is not already in use by another device.

The client can receive multiple DHCPOFFER packets from multiple DHCP servers located in the same segment.

The client selects one IP address and broadcasts a DHCPREQUEST packet to the server that owns the IP address that is going to be used. The decision depends on the period that the IP address can be borrowed or other information that the specific client needs for optimal operation. The rest of DHCP servers should listen to the DHCPREQUEST sent to the chosen DHCP server so that they continue with their normal functioning.

The chosen server returns a DHCPACK indicating to the DHCP client that the IP address assignment is finalized. If the offer is no longer valid for some reason (e.g. time-out or another client had allocated the same IP address) the chosen server

must respond with a DHCPNAK message. Upon receiving the DHCPNAK, the client will initiate the procedure and will send another DHCPDISCOVER packet.
The client receives a DHCPACK and knows that it owns the IP address. However, the client may refuse the offer indicated in the DHCPACK message. The clients can use ARP messages to check that the offered IP address is not in use. In that case another node would respond to the ARP message the client would assume that the offered address is in use and will refuse the offer by sending a DHCPDECLINE message to the DHCP server. If this happens, the client will re-initiate the procedure and will send another DHCPDISCOVER packet.

The client has to renew the IP address assignment before it expires by sending another DHCPREQUEST message. If the client does not need the IP address anymore, it will send a DHCPRELEASE message to the DHCP server so that the IP address is available for other nodes. If the server does not receive a DHCPREQUEST from the client before the IP address assignment expires, the server marks the lease of the address as non-renewed and makes it available for other clients. Having a service discovery mechanism that provides information to the Ad Hoc nodes about DHCP servers will enable connectivity between Ad Hoc and fixed IP networks. Therefore, the Ad Hoc node can discover the existence of a DHCP server and it can request a public address to communicate with fixed infrastructure.

If a DHCP server is available, it means that the network is managed and the Ad Hoc node should use the assigned IP address. If no DHCP server is available, it means that the network is unmanaged or Ad Hoc. In this case the device should discover the IP address itself (e.g. use Auto IP to get an address). Auto IP defines how a device chooses an IP address from a set of reserved addresses. This feature is already inbuilt in IPv6 but since IPv4 networks will last for a certain time, this thesis will consider IPv4. Auto-IP defines the algorithm for choosing an address from the 169.254/16 address range. After selecting the IP address, the Ad Hoc node has to test it to determine if the address is already in use. If the address is in use by

another device, the node has to pick another address randomly and test it again. This process can be repeated a few times and if it fails then the node cannot have connectivity.

The node has to test the selected address using an Address Resolution Protocol (ARP) probe. An ARP probe is an ARP request with the device hardware address used as the sender's hardware address and the sender's IP address set to 0s. The device will then listen for responses to the ARP probe, or other ARP probes for the same IP address. If the Ad Hoc node receives any of these ARP packets, the node must consider the address is in use and it should try a different address.

2.1.2.2 Address Resolution Protocol

The Address Resolution Protocol (ARP) [21] is a protocol used by the Internet Protocol (IP) [22] for mapping network layer onto the Ethernet or other transport. The ARP protocol maps IP network addresses to the hardware addresses used by the data link protocol. The ARP protocol operates below the network layer as a part of the OSI link layer. It implements the address resolution, which consists of finding an address of a computer in a network.

Ethernet networks include the hardware addresses to identify the source and destination for each packet sent over the network. Ethernet standards name the hardware address as the Medium Access Control (MAC) address (e.g. destination MAC address with all 1's identify a broadcast packet, which is sent to all devices connected in the Ethernet network). The Ethernet MAC addresses are globally unique (6 bytes) and are stamped in the network interface when they are manufactured.

IP protocol operates at the network layer and uses its own addressing space and ARP protocol performs the address resolution between both address spaces (i.e. Ethernet at data link layer and IP at network layer). The ARP protocol client and server are normally integrated with the driver included in the network interface card. The ARP protocol consists of four messages differentiated by the "operation"

field; ARP request, ARP reply, RARP request and RARP reply. The ARP client maintains a cache with the most recently resolved addresses.

In the normal functioning of ARP protocol, a computer sends all packets with its own hardware source address, and receives all packets, which match its hardware address or the broadcast address.

When the IP address of the destination node is known, the local node issues an ARP request message to find the MAC address of the destination node. The ARP request consists of a broadcast message containing the IP and MAC addresses of the local node and the IP address of the destination node. The message type in the ARP request includes the Ethernet protocol code (i.e. 0x806). All the nodes will discard the packet except the target node, which will send a unicast ARP response message to the originating node. The ARP response will contain the destination node MAC address. If the IP address does not belong to the Ethernet segment the MAC address cannot be resolved and an error will be returned.

After successfully configuring an Auto-IP address the node obtains an IP address but still in order to gain network layer connectivity, the address has to be has to be visible from external networks in order to receive messages from anybody, and the address has to be routable to send messages towards external networks.

In order to receive messages from external networks the new IP address has to be visible (e.g. Mobile IP). In order to send messages towards external networks the address has to be shared among routers by the existing routing protocols (e.g. RIP, OSPF, BGP, etc).

2.1.2.3 Routing Information Protocol

The Routing Information Protocol (RIP) is one of the oldest and most mature routing protocols within the IP family. RIP is defined in Request For Comments (RFC) 1058 [8] and Internet Standard (STD) 56. However, as the IP networks are

growing dramatically the Internet Engineering Task Force (IETF) [23] has done some updates in RFC 1388 [24] and RFC 1723 [25], named as RIP 2. Despite these updates, the original RIP still remains as a non-obsolete protocol. RIP 2 includes additional information identified as necessary in the new IP networks such as support for subnet masks and authentication for securing the table updates.

The success of RIP may be due to its simplicity. RIP sends routing-update messages at regular intervals or when there is a topology change. The routers receive the updates and refresh the entries in their routing table. After updating the routing table, the router sends updates to inform other routers about the changes. When the router sends those updates, it increases the metric value for the path by 1, and the sender is indicated as the next hop. The routers maintain only the best route, which is calculated as the route with the lowest metric to a destination.
RIP protocol limits the number of hops to 15 for preventing routing loops to continue indefinitely. Thus, if a router receives a metric value and after increasing it reaches the infinity (i.e. metric value equals to 16), the router will consider the destination as unreachable.

RIP protocol includes other stability features despite potentially rapid changes in a network's topology (e.g. split horizon and hold down mechanisms to prevent incorrect routing information from being propagated, a routing-update timer, a route-timeout timer, and a route-flush timer to regulate its performance). Nevertheless, RIP is an old protocol that has showed some problems such as counting to infinity, loops and slow convergence.

2.1.2.4 Open Shortest Path First

The growth of voice and data services over IP generates significant scalability issues. This is the reason promoting the adoption of link state routing protocols such as Open Shortest Path First (OSPF, RFC 1247 [9]) to become a de facto routing protocol replacing the early distance vector protocols like RIP [8]. RIP is limited and simple while OSPF is very powerful but complex. After moving forward with OSPF replacing RIP, it seems that OSPF also will face the need for

some changes or enhancements. OSPF was derived from the Shortest Path First (SPF) algorithm also referred as the Dijkstra algorithm developed in 1978 for the ARPANET and an early version for the OSI's Intermediate System-to-Intermediate System (IS-IS) routing protocol.

OSPF is a link-state routing protocol that sends Link-State Advertisements (LSAs) to other routers within the same hierarchical area. The LSAs include information about attached interfaces, metrics used, and other variables. The routers use the SPF algorithm to calculate the shortest path to each node based on the information received in the LSA messages. Routing protocols such as RIP and IGRP are distance vector protocols that send all or a portion of their routing tables on each routing update. Table 2 shows an example of routing information stored in an OSPF server.

Table 2. An OSPF link state database.

Type	ID	Adv Rtr	Seq	Age	Cksum	Len
Router	10.250.240.8	10.250.240.8	0x800001fc	2388	0x3684	36
Router	10.250.240.17	10.250.240.17	0x80000217	1835	0x444c	36
Router	10.250.240.32	10.250.240.32	0x80000232	1876	0x0158	36
Router	10.250.240.35	10.250.240.35	0x80000291	1100	0x4aa5	36
Network	192.168.254.230	10.250.240.8	0x800001cc	117	0xab67	40
Summary	10.1.2.0	10.250.240.17	0x80000216	1535	0x1729	28
Summary	10.1.3.34	10.250.240.8	0x8000013a	2217	0x842f	28
Summary	10.1.3.34	10.250.240.35	0x800001a3	800	0x0f20	28
Summary	10.1.3.35	10.250.240.8	0x8000013b	288	0x7839	28
Summary	10.1.3.36	10.250.240.35	0x800001a4	1430	0xf833	28
Summary	10.1.2.48	10.250.240.8	0x800001f7	1317	0xaa8d	28
Summary	10.1.2.48	10.250.240.32	0x80000232	430	0xa242	28

Moreover, OSPF can operate within a hierarchy. The largest hierarchical entity is the Autonomous System (AS) that consists of a collection of networks under the same administrator. An AS can be divided into a number of areas. The routers with multiple interfaces (i.e. Area Border Routers) can participate in multiple areas by keeping separate topological databases for each area, and using multiple instances of OSPF as the intra-AS routing protocol.

The AS border routers running OSPF learn about exterior routes through exterior gateway protocols, such as Exterior Gateway Protocol (EGP) and Border Gateway Protocol (BGP), or through static configuration information.

2.1.2.5 Border Gateway Protocol

In the early stages Internet was limited to a set of local networks directly connected by routers or "gateways". When the network grew it was necessary to adopt a hierarchical structure and split the network into a set of Autonomous Systems (AS). The Exterior Gateway Protocol (EGP) [26] was the selected protocol for exchanging routing information between ASs. Similarly to what happened with RIP and OSPF, the Internet kept growing and the EGP presented a set of limitations that led to the adoption of the Border Gateway Protocol (BGP) [20]. The BGP solved the backbone-centered tree topology problem set by EGP. EGP was running as an AS-level distance vector protocol and it had a set of limitations that BGP overcame. Thus, in both cases of intra-domain and inter-domain routing protocols, the constant growth of the network forced the adoption of new protocols such as OSPF and BGP. Therefore, in the situation where the Internet is expected to grow considerably with the introduction of IPv6, a set of enhancements and improvements such as security and scalability are required. BGP is used to exchange routing information between ASs like the ones formed by the Internet Service Providers (ISP). Figure 3 shows the network architecture with OSPF and BGP protocols distributing the intra and inter-domain routing information.

Internally the networks (e.g. universities and corporations) use Interior Gateway Protocols (IGP) such as RIP or OSPF for exchanging the routing information.

There are different flavors of BGP depending whether the protocol is used for exchanging routing information between autonomous systems (AS) (i.e. BGP is referred to as External BGP: EBGP) or BGP is used for exchanging routes within an AS (i.e. BGP is referred as Interior BGP: IBGP).

BGP is very robust and maintains considerable scalability by using route parameters (i.e. attributes) to define routing policies and classless interdomain routing (CIDR) for reducing the size of the routing tables (e.g. normally BGP routing tables maintain more than 90,000 routes and in the non default routers that store the routes to all destinations, the entries are in the order of 130,000 routes).

BGP routers do not send periodic routing updates. Instead only when a BGP router detects a change on its routing table it will exchange routing information including only these routes that have changed. BGP routers establish TCP connections between themselves for exchanging the routing updates advertising only the optimal path to a destination network. BGP uses a set of properties (i.e. BGP attributes) to determine the best route to a destination when the routing table contains multiple paths to a particular destination.

Therefore, one possibility for Ad Hoc nodes to have IP addresses routable towards fixed infrastructure is to include the IP address into the OSPF and BGP tables. This will allow the Ad Hoc nodes to be reachable from the global Internet. However, the Ad Hoc nodes may obtain the IP address from the non-routable address range using Auto-IP process. Moreover, the Ad Hoc nodes change dynamically the point of attachment to the network. Thus, Ad hoc nodes IP addresses should not be visible in BGP since it means a lot of updates every time the node changes the IP address and it has to be updated in the BGP tables. Scalability of BGP is a serious problem in the actual Internet. BGP cannot tolerate frequent appearance and disappearance of nodes like in Ad Hoc networks.

The alternative would be that Ad Hoc node get connected to the fixed wireless infrastructure via a Gateway that includes Network Address Translation (NAT) functionality.

2.1.2.6 Network Address Translation

Basic Network Address Translation or Basic NAT is a method by which IP addresses are mapped from one group to another, transparent to end users. The need for IP Address translation arises when a network's internal IP addresses cannot be used outside the network either for privacy reasons or because they are invalid for use outside the network. Basic Address translation would allow hosts in a private network to transparently access the external network and enable access to local hosts from the outside. There are limitations to using the translation method. It is mandatory that all requests and responses pertaining to a session be routed via the same NAT router. One way to ascertain this would be to have NAT based on a border router that is unique to a stub domain, where all IP packets are either originated from the domain or destined to the domain.

Having a NAT between the Ad Hoc and the fixed networks allows the Ad Hoc nodes to move and change their IP address dynamically. The NAT will provide the address mapping when the Ad Hoc nodes want to communicate with another device in the fixed infrastructure as presented in Figure 5.

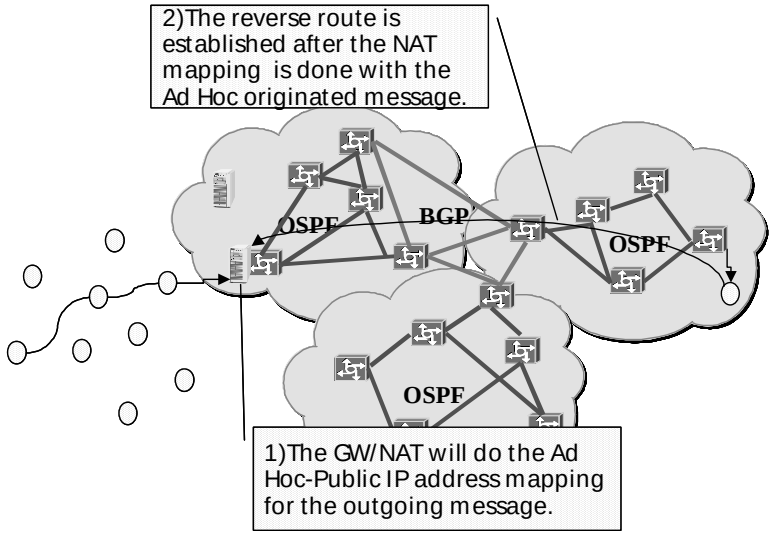

Figure 5. Ad Hoc network attached to fixed infrastructure via GW with NAT functionality.

The Ad Hoc node will find the Gateway to connect to the fixed infrastructure and from there the NAT will do the mapping. This allows Ad Hoc nodes to be visible

to external nodes by publishing in the BGP table the static address (i.e. the address of the GW) and only the prefix maintained by the NAT.

The server acting simultaneously as an Ad Hoc node and a bridge (i.e. GW with NAT functionality) to the fixed network, should include in the BGP routing tables the network address under the administrative domain of the Gateway to provide connectivity to all the Ad Hoc nodes located behind the Gateway in the Ad Hoc network.

This allows a mobile originated connectivity that will trigger the mapping into public IP address at the Ad Hoc gateway. The Mobile terminated is not feasible since the Ad Hoc node is not visible outside unless the node maintains the old IP address but registers the new IP address acquired in the Ad Hoc network using Mobile IP.

2.1.2.6 Mobile IP

Mobile IP allows a device to maintain the same IP address (its home address) wherever it attaches to a new network. The mobile device has a care-of address, which relates to the subnet where it is currently located. A home agent manages the care-of address. The Home agent is a device on the home subnet of the mobile device. Any packet addressed to the IP address of the mobile device is intercepted by the home agent and then forwarded on to the care-of address through a tunnel. Once it arrives at the end of the tunnel, the packet is delivered to the mobile device. The mobile node generally uses its home address as the source address of all packets that it sends.

The following terminology is used in a mobile IP:

Home Address. The static IP address allocated to a mobile node. It does not change no matter where the node attaches to the network.

Home Network. A subnet with a network prefix matching the home address of the mobile node. Packets intended for the home address of the mobile node will always be routed to this network.

Tunnel. The path followed by an encapsulated packet.

Visited Network. A network to which the mobile node is connected, other than the node's home network.

Home Agent. A router on the home network of the mobile node that maintains current location information for the node and tunnels packet for delivery to the node when it is away from home.

Foreign Agent. A router on a visited network that registers the presence of a mobile node and detunnels and forwards packet to the node that have been tunneled by the mobile node's home agent.

Mobile IP process in Ad Hoc networks

The Mobility agents (home agents and foreign agents) advertise their presence on the network by means of agent advertisement messages, which are ICMP router advertisement messages with extensions. A mobile node may also explicitly request one of these messages with an agent solicitation message. When a mobile node connects to the Ad Hoc network may receive one of these messages then it follows the normal procedure. The node detects from an agent advertisement that it has moved to a foreign network, and can obtain a care-of address for the foreign network. After mobile node has received its care-of address, it needs to register itself with its home agent. This may be done through the foreign agent, which forwards the request to the home agent, or directly with the home agent. These communications between a mobile node and a foreign agent takes place at the link layer level. It cannot use the normal IP routing mechanism, because the mobile node's IP address does not belong to the subnet in which it is currently located. Once the home agent has registered the care-of address for the mobile node in its new position, any datagram intended for the home address of the mobile node is intercepted by the home agent and tunneled to the care-of address. The tunnel endpoint may be at a foreign agent if the mobile node has a foreign agent care-of address), or at the mobile node itself (if it has a co-located care-of address). Here the original datagram is removed from the tunnel and delivered to the mobile node.

The mobile node will generally respond to the received packet using standard IP routing mechanisms as described in .

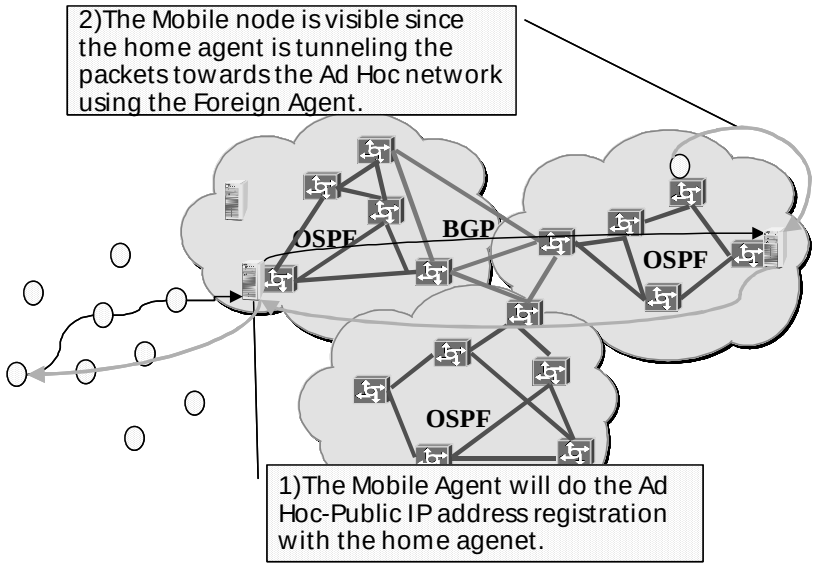

Figure 6. Mobile IP within the Ad Hoc context.

In the scenario where the mobile node did not receive any agent advertisement it has to find any available mobile agent to request a care of address and register and obtain connectivity at network layer. This process can be done in Ad Hoc networks using the normal procedure where the mobile node broadcast the query to find mobile agent. An alternative procedure consists of having the information about existing mobile agents already available in some Ad Hoc nodes. Thus, when the node attaches to the network, it immediately obtains the mobile agent address as part of the routing data.

Thus, for mobile terminated the connectivity would be provided at network layer using Mobile IP and session layer as described in chapter 6.

Therefore, in order to enable the network layer connectivity, the Ad Hoc gateway or the Mobile Agent has to be found among all the Ad Hoc nodes, and a service discovery is required. However, the existing service discovery mechanisms implemented at application layer do not suit for AD Hoc networks. In this case a service discovery solution that helps at network and session layer would solve the

Ad Hoc connectivity problem. In the next few sections we will discuss the service discovery mechanisms implemented nowadays on existing fixed IP networks.

2.1.3 Service discovery in fixed IP networks

In addition to the protocols defined for routing, addressing and name translation as described in previous sections, the IP networks have become a technology breakthrough because of the inbuilt plug in capability to create new services such as Web, E-mail, File Transfer, Content sharing, among others. Moreover, IP networks have been extended also to support real time services such as media provisioning, streaming and Voice over IP (VoIP). Each of these services requires its own service management protocol such as HTTP, SMTP, FTP, H.323, SIP, etc. However, IP networks provide an easy mechanism to plug and develop new services that are accessible from any IP endpoint. Services can be considered, as an intangible product consisting of activities between the service provider and the user to meet the user needs. A service includes a set of functionality that fulfills the user needs. The services can be software components that employ different technologies. For example, in circuit switched or even in IP telephony, call forwarding, call waiting [60] were considered as call related services. In IP based networks, Web Services (e.g. using HTTP as transport together with SOAP, WSDL and UDDI), messaging, presence, file transfer and location based services can be provided in a distributed or centralized manner by application servers.

All those services can be reachable anywhere at anytime, thus a set of application layer protocols for finding the services available in the network is required. These protocols solve the service discovery problem in IP networks where services may include a huge variety of functionality and can be located anywhere. There are different proposals to implement service discovery. This section presents and evaluates some of them (Bluetooth-SDP, JINI, UPnP, Salutation, OSGi, SLP, etc).

2.1.3.1 The Bluetooth Service Discovery Protocol (SDP)

The Service Discovery Protocol (SDP) is a solution defined as part of the Bluetooth specifications [68]. The number of services that can be provided over

Bluetooth links may increase in the near future. Therefore, Bluetooth specifications define a set of procedures to enable the Bluetooth devices to discover the variety of services available. The Bluetooth protocol has defined the SDP as part of the stack to be used by any service discovery application to locate the services available. The service discovery application does not make use of SDP protocol as means for accessing a service, but rather for informing the user about the services that are available to his/her device. Thus all Bluetooth applications running in a local device can use the SDP procedure to retrieve any pertinent information that will facilitate the application to access the selected service in a remote device. SDP provides an abstraction layer with a set of primitives that any application in the device can use to search an existing service.

SDP uses the Connection-Oriented (CO) transport service from the Link Layer (L2CAP), which in turn uses the base band Asynchronous Connectionless (ACL) Links to ultimately carry the SDP Protocol Data Units (PDU) over the air. SDP is based on the Client-Server query response model. An SDP server contains a service records database that is the repository of service discovery-related information.

During the configuration process for the service discovery mechanism a set of roles such as Local device and Remote Device are defined. The former is the device that initiates the service discovery procedure. It contains the "client" portion of the SDP architecture and the service discovery application used by a user to initiate the discoveries and display the results. The latter is any device that participates in the discovery procedure and responds to the service enquiries generated by the Local Device. It must contain the "server" portion of the SDP architecture and the record database to consult and create a response.

Thus when the Bluetooth devices are powered-on and initialized, a link is created by a discovery inquiry process and paging to the other device. The piconet infrastructure is established and either a master or slave can initiate the discovery

process. Afterwards, the user will have a complete knowledge of the services available in the piconet (Ad Hoc network).

The advantages of the Service Discovery Protocol

The SDP protocol performs an in-built mechanism that can be used by any Bluetooth-enabled device. SDP as part of the Bluetooth stack is tightly linked to the basic link layer functioning. SDP provides a reliable and robust mechanism for auto-configuration of any Ad Hoc network built on top of Bluetooth devices. SDP is flexible in the sense that it can announce any type of service available in the network and furthermore it can be extended to new services. It can also support other service discovery mechanisms that will use the primitives abstract layer defined in SDP. The service discovery application can rely on the SDP primitives for its functionality when it is working on top of Bluetooth devices, or it can use other mechanisms defined by other technologies (HAVi, etc.)

The drawbacks of the Service Discovery Protocol

This mechanism has the inconvenience that it is specific to Bluetooth devices and it cannot be used in a general manner for any Ad Hoc network. The philosophy of Ad Hoc network is that it is built by a diversity of devices. Therefore, SDP suits for Bluetooth based networks but it cannot be extended to other technologies without changes. The optimal solution would be one that can be used independently of the technology and still keep the design requirements presented in SDP.

2.1.3.2 Jini™ network technology

A Jini [46] system is a distributed system with the aim of converting the network into an easily administered system, where clients (humans or devices) can find the necessary resources. The resources can be considered as hardware devices or software programs. The main goal is to make the network a dynamic entity that enables to add and delete services and make them available in a seamless manner.

The end goals of the JINI technology include the following:

- Provide easy access for the users to access the services and resources in the network.

- Enable mobility of the user attachment to the network and provide the users easy access to the resources as they move.
- Provide the appropriate dynamics to network configuration by simplifying the maintenance and management of network devices, software, and users as they change.

The JINI system is based on the Java application environment and therefore it provides a distributed computing platform because of the code mobility. This approach allows a smooth configuration of resources across the network.
The JINI technology provides the infrastructure for devices, services, and users to be incorporated into the network and detach from it in a smooth manner. These assumptions rely on a good speed connection among network devices and a reasonable latency all over the network.

The main concept in the JINI technology resides in the fact that a service is considered as an entity (i.e. a person, a program, or another service, storage or a communication channel). Those services are found using a lookup mechanism and a service protocol is used for communicating with each other. This service protocol is implemented as a set of interfaces written in the Java programming language. The lookup mechanism maps the interfaces that provide the service functionality with the objects that implement the service itself. The lookup mechanism contains a couple of protocols ("discovery" and "join") for adding any new service to the lookup service. All these services communicate using the Java Remote Method Invocation (RMI) that is a traditional Remote Procedure Call (RPC) mechanism.

Jini technology drawbacks

This alternative has some drawbacks. The Code mobility (JINI) appears as an efficient solution for supporting information retrieval. The problem is that in an environment in which bandwidth is a scarce resource and users' mobility makes continuous communication, the code mobility is very complex (if not impossible). Furthermore, the nodes should be designed in such a way that the peers accept the others' code. Since code mobility gives to the users the access to other machines,

security is a concern and the literature hardly addresses the security of Ad Hoc networks.

2.1.3.3 The Universal Plug and Play (UPnP)

The Universal Plug and Play Forum [47] is an industry initiative for enabling multivendor connectivity among wireless devices, intelligent appliances and PCs. The objective is to define and publish UPnP device and service descriptions, and the members of the UPnP Forum can create the means to easily connect devices and simplify the implementation of networks.

The UPnP architecture consists of a TCP/IP communications channel and Web services that are used for transferring data among the multiple devices. These devices describe their capabilities and features when they join the network using the UPnP protocols (device addressing, device discovery, device description, action invocation, event messaging and presentation or human interface). Therefore, when new devices join the network, they can receive the description of existing devices and those new devices can use that description without a complex configuration.

UPnP device discovery

The UPnP architecture consists of a control point that queries the network devices for checking their status and their service description, including the URL that addresses the device that implements the service. This is a similar functionality to the JINI lookup mechanism. The UPnP devices communicate with the control point using TCP/IP. The role of the control point is assigned to any device on the network. Thus, when a new device joins the network, it has to announce its capabilities and service characteristics to all control points to be aware about the new active device on the network. The control point can have an active role and search for new devices that have joined the network. In case the control point discovers a new device, the control point has to request a service description that includes the services it provides. The service description is formatted in an XML document. UPnP specifications define the bootstrap mechanism where the devices have to check whether there is a DHCP server available from where they can obtain their addresses dynamically. If there are not DHCP servers available the

nodes can use “Auto IP” [89], [85]. The “Auto IP” consists of an automatic mechanism where the UPnP device selects an IP address randomly (from a range of non-routable addresses) and tests it to ensure that no other device is using it.

UPnP drawbacks

UPnP is a fully distributed service discovery protocol that suits for Ad Hoc networks. UPnP includes Auto IP address assignment and uses well-established protocols such as HTTP, SOAP, etc. However, UPnP does not scale for medium to large networks and exhaust the nodes energy since the UPnP architecture is based on continuous multicast service announcement over the network. Therefore, the nodes exhaust their battery when keeping refreshing their services and the network gets flooded with the announcements.

2.1.3.4 The Salutation protocol

Even though JINI and Bluetooth both have a discovery protocol, the two protocols are incompatible. A JINI device cannot communicate with a phone that uses only the Bluetooth service discovery protocol. However, Salutation [48], the discovery protocol developed by the Salutation Consortium, may turn out to be the bridge between Bluetooth and JINI devices, as well as to other platforms. Were the devices to use the same protocol, they would at least be able to determine the capabilities of the other devices in the network. What's more, because the same software could be used with a Java- or Bluetooth-enabled device, or one using some other network protocol, Salutation may be the most cost-effective way to develop applications for any networked device.

The Salutation Specification describes an API, a Protocol and optional data pipes and job control to assist in discovery and use of devices, applications and services that may encounter each other in a communication setting. The architecture is based on a model called the Salutation Manager. The Salutation architecture enables transactions between a fax machine and a copier, for example. The Salutation protocol supports the effort to integrate different devices into a network

by supplying them with a standard communications and API specification for discovering the capabilities of other entities in a network.

Figure 7, shows the overall architecture in the Salutation protocol. The server [left] uses the Salutation discovery protocol to ask other devices on its network about their capabilities. The inquiry passes from the Salutation manager to a transport manager, which prepares the inquiry to run over the transport protocol used by the network. It makes its way over the network to, for example, the device-server combination [centre], which supplies the information, and also learns about the server. The device-server combination is also connected to a different device by a second network running a different transport protocol [right]. The centre combination needs two transport managers, one for each network protocol, but only one Salutation protocol.

The Salutation application interface allows software programs, such as e-mail or word processing applications, to interact with the Salutation protocol. Bluetooth Special Interest Group (SIG) and the Salutation Consortium have paired off to synchronize their service discovery approaches, thus allowing software developers to write one application that works with Bluetooth and existing Salutation environments.

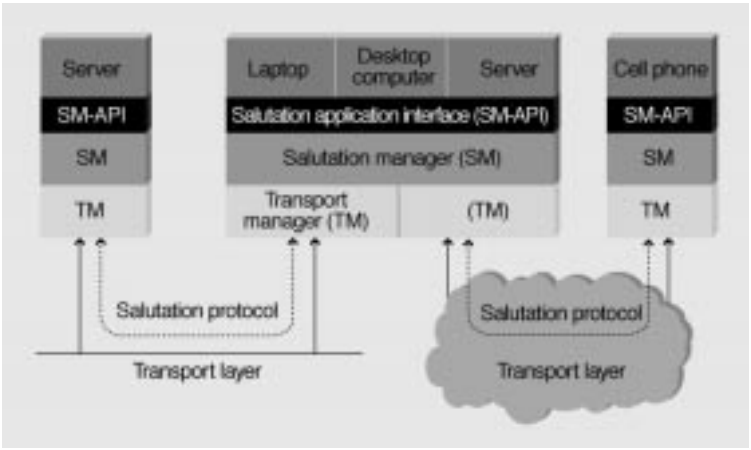

Figure 7. Salutation protocol architecture.

Salutation drawbacks

Similarly to JINI, Salutation technology results in quite heavy procedures for Ad Hoc networks. Therefore, Salutation does not suit for an environment in which bandwidth is a scarce resource and devices have to include routing functionality and self-management procedures. In such an environment, the service discovery should be light and suitable for devices with low resources.

2.1.3.5 The Open Services Gateway Initiative (OSGi)

The Open Services Gateway Initiative (OSGi) [49], established in 1999, is an independent, non-profit corporation working to define and promote open specifications for the delivery of managed broadband services to networks in homes, cars and other environments. The OSGi specification is designed to complement and enhance virtually all residential networking standards and initiatives. In the same way, the specification leverages the value of existing wire line and wireless networks while providing flexibility toward cable, WCDMA, xDSL and other high-speed access technologies.

The OSGi Framework and Specifications as depicted in Figure 8 facilitate the installation and operation of multiple services on a single Open Services Gateway (set-top box, cable or DSL modem, PC, Web phone, automotive, multimedia gateway or dedicated residential gateway).

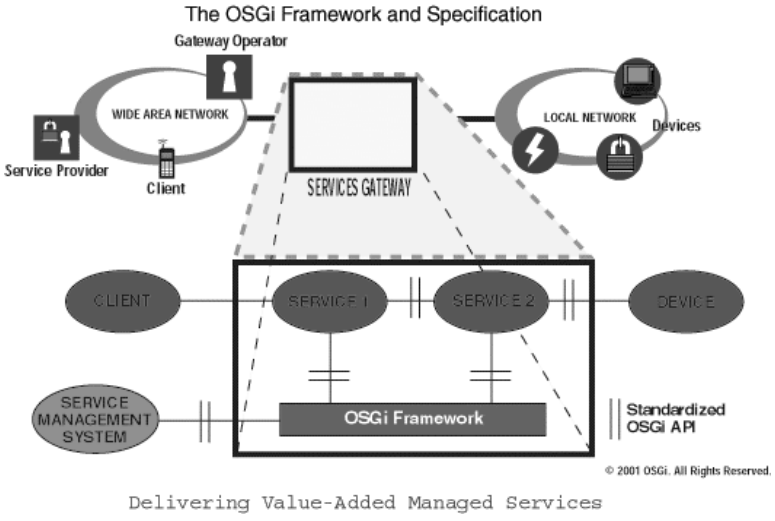

Figure 8. OSGi Gateway infrastructure.

The OSGi Specifications delineate Application Programming Interface (API) standards for a gateway platform execution environment. Open Services Gateways must support these API standards in order to conform to the OSGi specification. The APIs address service cradle-to-grave life cycle management, inter-service dependencies, data management, device management, client access, resource management and security. Using these APIs, end-users can load network-based services on demand from the Service Provider while the Gateway manages the installation, versioning and configuration of these services.

OSGi Drawbacks

The OSGi cannot be considered as service discovery protocol per se, it provides a framework for implementing gateway functionality to access available services in local networks (e.g. Home networks).

2.1.3.6 The Service Location Protocol (SLP)

The Service Location Protocol is an IETF protocol defined for service discovery. The Service Location Protocol (SLP) [50] provides a lightweight mechanism for service discovery within one administrative domain. In addition to the local domain, SLP can be used to discover desired services in specific remote DNS domains. The key issue for remote discovery in SLP is to enable a User Agent (UA) to learn about remote Directory Agents (DAs) or/and remote Service Agents (SAs) without relying on multicast.

SLP Drawbacks

The SLP is based on the knowledge of the SLP server where the clients can address the service request. Furthermore, the service provisioning has to register the service characteristics in the SLP server. Thus, when the client accesses the server, it will get all the service information in the Service reply. Therefore, SLP provides a centralized service discovery mechanism, which does not suit for Ad Hoc networks where the services and the service discovery have to be fully distributed.

2.2 Addressing and routing in wireless IP networks

In the above sections we described the approach and protocols used for the addressing and routing in IP networks. According to the Internet design philosophy based on building blocks approach, new bearers can be included, as part of IP networks. Thus, IP protocol patches together the different technologies without damaging the coherency of the overall architecture. This section presents the evolution of wireless networks to become part of the bearer technologies pulling IP networks out of wired environments. This section describes the mechanism that wireless with fixed infrastructure implement for supporting mobile nodes. The wireless networks have provide an IP layer with mobility support by having a radio layer that includes roaming and tunnels the IP packets into an specific wireless transport protocol (i.e. GTP).

2.2.1 2G-2.5G IP wireless networks

The 1G wireless networks introduced mobility features to the universal voice communications. The 1G networks were analogue systems lacking performance and efficiency. Nevertheless, 1G had already a huge impact and immediately a big effort was initiated in order to harmonize the wireless communications and the digital technology paved the process. Thus, 2G or GSM [35] networks were deployed all around Europe and later over much of the rest of the world. The 2G networks have been continuously evolving and have gone through different phases towards convergence with IP networks. The 2G wireless networks phase 1 (GSM) standards, specified all necessary components for mobile transmission of voice (i.e. rate: 9.6 kbps up to 13Kbps) and user data transmission (i.e. range from 0.3 kbps up to 9.6 kbps). The 2G wireless networks phase 2 included a set of supplementary services (e.g. half rate speech; 5.6 kbps). The GSM phase 2+ is the one that is named as 2.5G wireless networks. In this phase a set of features was introduced in order to increase the transmission data rates (e.g. High Speed Circuit Switched Data; HSCSD, Enhanced Data Rates for GSM Evolution; EDGE, General Packet Radio Service; GPRS, etc).

The HSCSD (High Speed Circuit Switched Data) allows the transmission of more than 100 kbps, using a circuit switched connection. HSCSD introduced a new codec that allow up to 14.4 kbps to be transmitted via one physical channel and new software upgrades that allow the possibility to bundle up to 8 physical channels (of one carrier).

The EDGE (Enhanced Data Rates for GSM Evolution) modifies the modulation technique on the air interface thus allowing a higher transmission rate on a physical channel. EDGE introduces the GMSK modulation (i.e. 8PSK) for increasing the transmission rate of up to 69.2 kbps per physical channel and the channel bundling that provides a transmission rate of up to 384 kbps.

The GPRS (General Packet Radio Service) is a packet switched service, which also allows transmission rates up to 171 kbps. GPRS introduces new network elements in order to provide IP connectivity to the public Internet and other IP networks.

The main objective in 2.5G [35] or GSM Phase 2+ was the creation of a platform that later on would be used for 3G or the 3rd generation mobile communication system UMTS. ITU is responsible for the standardization of the IMT-2000, which shall be the guideline to all 3rd generation mobile communication (3G) systems. The 2.5G wireless networks as shown in Figure 9, are an extension of 2G networks so they are based on Global System Mobile (GSM) standards with the difference that 2.5G provides support for establishing IP tunnels. In 2.5G networks applications can see an IP layer and underneath the radio technologies manage to provide a reliable pipe for data packets. In the 2.5G networks the terminals have a GSM radio interface and use additional GSM channels for establishing an IP channel. In order to increase the data rate, GPRS bundles multiple physical channels (e.g. up to 8 time slots of a TDMA-frame can be dynamically bundled for a single user).

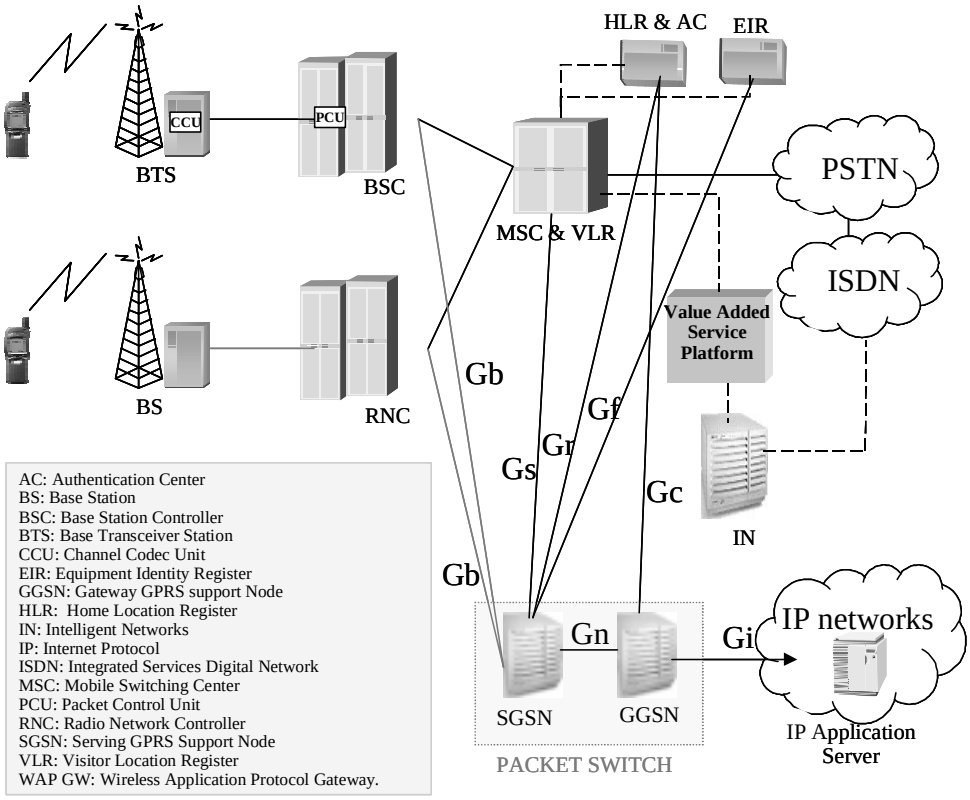

Figure 9. Network elements in the 2.5G infrastructure.

Thus, GPRS offers a more efficient use of the radio resources and the option to gradually move to the 3rd generation mobile communication systems. The evolution path from GPRS based on 2G networks (i.e. GSM) towards 3G [35] networks is guaranteed with the separation of the Network Switching Subsystem (NSS) and the Base Station Subsystem (BSS). This allows the BSS elements to evolve and they can be changed without affecting the NSS network elements. However, this separation requires new GPRS components to be added in the radio part (i.e. Packet Control Unit; PCU and Channel Codec Unit; CCU) and in the network side (i.e. Serving GPRS Support Node; SGSN), the Gateway GPRS Support Node; GGSN and the HLR extension).

The **Serving GPRS Support Node (SGSN)** is connected with the BSS via a Frame Relay Network and has to keep track of the location of the GPRS mobiles and perform security functions, access control to the network, etc.

The **Gateway GPRS Support Node (GGSN)** is connected with the fixed IP networks and has to implement the interworking between the PLMN NSS and external packet-switched networks.

The **HLR-Extension** maintains the subscriber data similarly to the GSM Home Location Register (HLR).

The **Packet Control Unit (PCU)** is responsible for the radio channel management, segmentation and re-assembly of packet data units (PDU), packet data channels (PDCH) scheduling and the channel access.

The **Channel Codec Unit (CCU)** is located in the BTS and performs the Channel Coding (including the coding scheme algorithms) and the power control procedures.

The GPRS network encapsulates the Packet Data units (PDU) received from the terminal, adds the required addressing information so Service Data Units (SDU) are created to be transparently exchanged across the network. The NSS is kept independent from the BSS and the radio interface via the Gb interface. This facilitates the evolution to UMTS. The operator preserves the core components (SGSN and GGSN) and may replace the GSM-BSS by the corresponding UMTS-BSS. The GPRS architecture defines a logical connection between the GPRS-MS and the SGSN using the SNDCP (SubNetwork Dependent Convergence Protocol) protocol layer. The LLC (Logical Link Control) is underneath the SNDCP and provides an interface independent of the used radio subsystem and medium access technique for routing SNDCP PDUs transparently through the BSS. Moreover, within the NSS architecture, the GPRS Support Nodes (GSNs) either SGSN or GGSN exchange different Packet Data Protocols (PDPs) encapsulated by the GPRS Tunneling Protocol (GTP). Figure 10 presents the whole GPRS architecture and the different protocols required for encapsulating the IP packets for crossing the GPRS nodes and reaching the remote device.

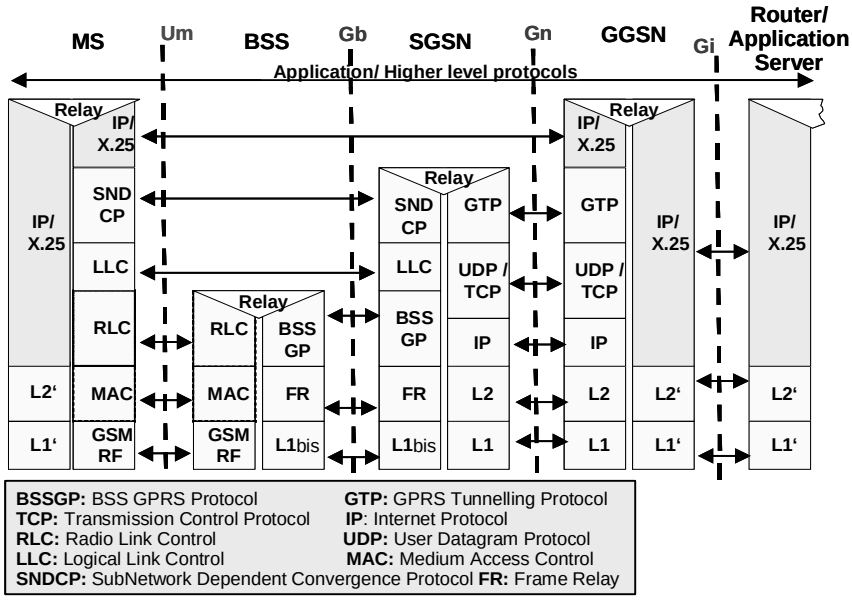

Figure 10. GPRS transmission planes.

The GTP PDU header contains the GSN addresses necessary for routing the tunnelling packets. The GTP header also includes the Tunnel Identifier (TID) to uniquely identify the GSN PDP context, which contains all information related to the MS and the GSN entities bound to the data channel opened for exchanging the IP data. The SGSN and MS exchange the N-PDUs packets encapsulated with the Subnetwork Dependent Convergence Protocol (SNDCP). The PDP context is uniquely identified with a Temporary Logical Link Identity (TLLI) and Network Layer Service Access Point Identifier (NSAPI) pair.

Figure 11 shows an example of PDP attach and activation procedure, required in order to set up an IP channel between the terminal (i.e. Mobile Station; MS) and the remote application server located at the external PDN providing a certain service (e.g. web browsing, e-mail access, etc). During the PDP attach and activation, the MS has to exchange a set of information with the SGSN. In this procedure the MS has to provide the NSAPI, the PDP Type (i.e. IPv4, X.25, etc), PDP Address (if the MS has been assigned a static IP address but if the field is left empty, then a dynamic address will be assigned), Access Point Name (i.e. a logical name is used to indicate the external PDN the subscriber wants to be connected to), QoS Requested (i.e. the desired QoS profile) and the PDP Configuration Options.

The SGSN has to perform the security functions and update the MS location in the HLR. The SGSN has also to determine the GGSN and construct the TID for the PDP created for the MS. The PDP context (i.e. including PDP Address, Access Point Name, QoS Negotiated, TID, Selection Mode and PDP Configuration Options) is sent to the GGSN. After this procedure the MS can start sending IP packets that will be encapsulated into the PDP PDUs and transferred between the MS and the GGSN that will extract the IP packets from the GTP and forward them to the remote PDN (i.e. Application server or other IP entity).

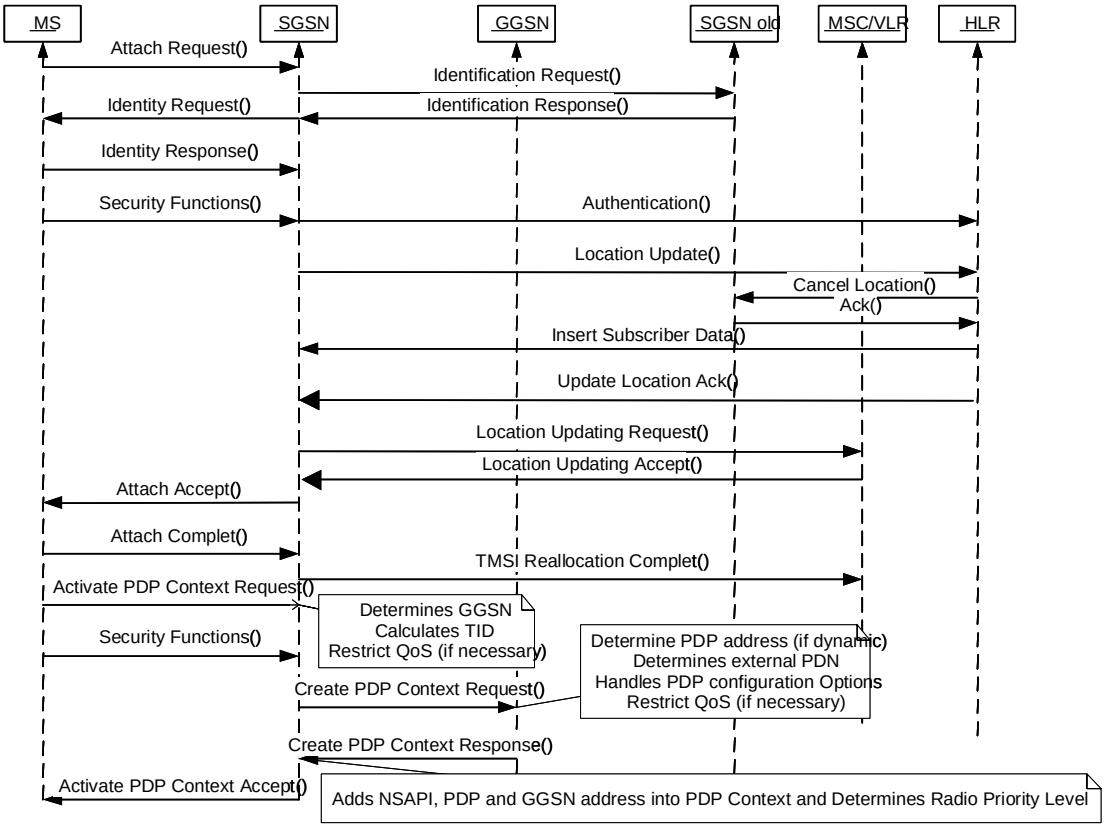

Figure 11. PDP attach and activation procedure.

Figure 11 shows the GPRS mechanism for providing connectivity at network layer to mobile nodes. During the attach procedure the nodes establish a tunneling layer built on top of the existign radio bearers. Therefore, the nodes obtain an transparent IP layer while the radio bearers perform the roaming and handover.

2.2.2 3G wireless networks

The 3G networks [35] are considered the next important step in wireless networks since they provide a completely new radio technology (i.e. WCDMA) and also the core network is based on fully IP based technology while remaining the old NSS. The 3G networks provide a new mobility model where in addition to the tunneling provided by the radio bearer, a new roaming mechanism is provided at session layer based on SIP. Further details are provided in chapter 6 about the mobility capabilities inbuilt in the SIP protocol.

The 2.5G wireless networks provide the mechanism for obtaining a data pipe for accessing IP networks. In 3G networks new network entities are introduced in order to provide Telephony or multimedia sessions set up functionality within IP based networks but also with legacy Circuit Switched networks.

The 3G networks are based on 2.5G networks infrastructure for contacting with the SGSN and GGSN in order to establish the IP channel. In addition to the basic GPRS functionality, the 3G terminals will have an additional radio interface based on WCDMA technology. The 3G networks are IP based and they use SIP as the signaling protocol. SIP is an application layer signaling protocol built on top of TCP/UDP/IP. IP is used as the network layer to communicate among all 3G network entities and to interact also with legacy entities such as the MSC and the HLR. The IP based network elements defined in 3G networks that are located after the GGSN (i.e. behind Gi Interface) form the so called IP Multimedia System (IMS). The IMS system, as shown in Figure 12, specifies a set of network entities such as Call Processing Server (CPS), Home Subscriber Server (HSS), and others that resemble the 2G MSC and HLR elements.

The Call Processing Server (CPS) is the network element that controls the Call State and performs the right decisions and handles the routing process to set up the call. The Call Processing Server (CPS) implements the Call State Control Function (CSCF) as part of the core network access control. CSCF is the logical entity that contains the SIP functionality for terminating the user-to-network signaling used

for network access. The CSCF, as part of the CPS, contains the SIP functionality of a proxy server, redirect and registrar described in the SIP protocol section. Depending on the precise functionality of the CSCF, UMTS specifications define another terminology. Thus, I-CSCF stands for Interrogating CSCF, which is the CSCF to which other CSCFs can query to obtain the callee address for routing to reach the end terminal. The S-CSCF denotes the Serving CSCF that after the terminal registration is the CSCF that will attend all the messages coming from and to the subscriber. P-CSCF is named the Proxy CSCF and it is just an intermediate CSCF that receives the incoming messages from the terminal and forwards them to the S-CSCF that attends that user. Other network entities required for interoperability with SCN networks are the Media Gateway (MGW), the Media Gateway Control Function (MGCF), and the Home Subscriber Server (HSS). The MGW converts the media stream from/to SCN to/from IP. The MGCF translates 3G SIP signaling to SCN. The HSS stores user profiles; it has the same functionality than the Home Location Register (HLR) in 2G networks. The HSS keeps user profiles and location information in addition to the original HLR and VLR that are used by the SGSN when opening the PDP context to acquire an IP pipe. HSS keeps the user information regarding service profile, terminal capabilities and so on. In addition to the security included in GPRS access networks, the IMS network includes security mechanisms at the application layer. The SIP [36] protocol implements a security handshake similar to the one performed during the PDP context attach and activation.

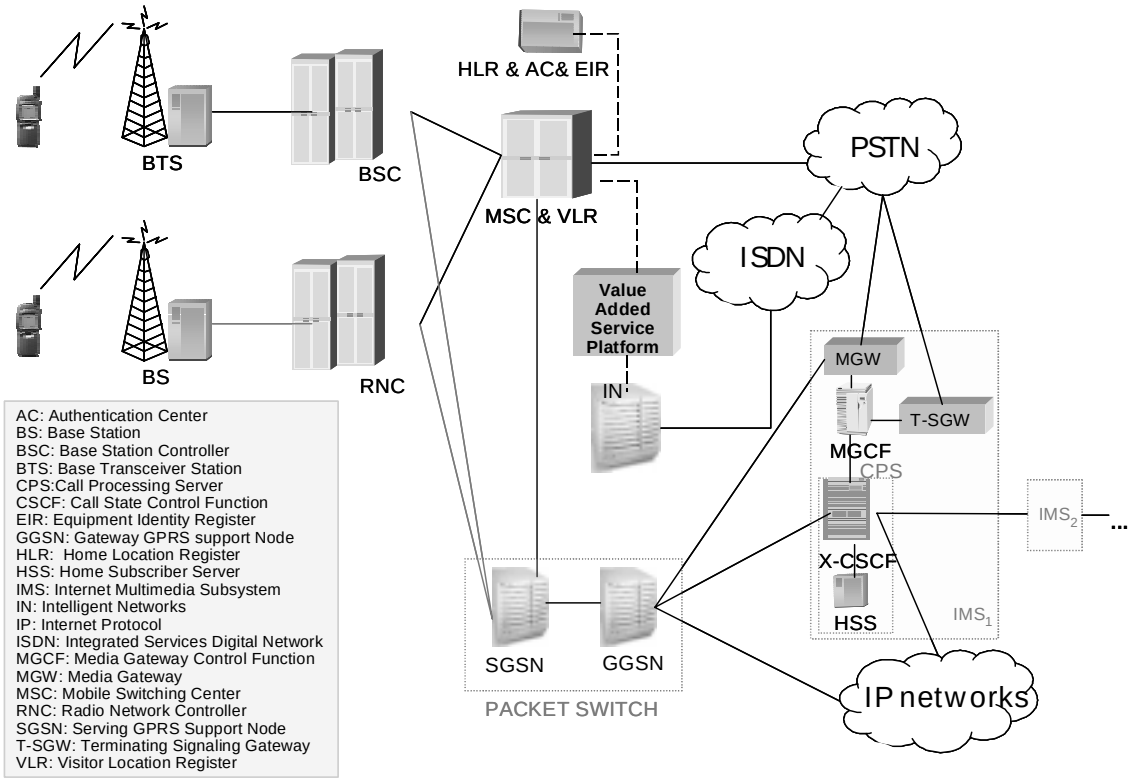

Figure 12. Network elements in 3G infrastructures.

2.2.3 WLAN networks

The wireless networks based on the 802.11 standard [5] are the other alternative for providing wireless IP connectivity. The bandwidth and the range that can be reached with this technology are the main differences comparing with 3G networks or other short-range technologies such as Bluetooth. The specification work of the standard IEEE 802.11 was started in 1990 and completed in 1997. The first phase of the standard IEEE 802.11 supports only 1 Mbit/s and 2 Mbit/s data rates. The following phase introduced the extension IEEE 802.11b that supports data rates of up to 11 Mbit/s with the radio frequency technology of direct spread spectrum sequence. WLAN uses the unlicensed frequency band between 2,4-2,483 GHz. The IEEE 802.11 wireless LAN is a local network and the coverage area forms small islands. The association or network attachment is the basic service that enables the connection between the station (STA) and the Access Point (AP) in a WLAN

working in the infrastructure mode. The wireless LAN specifications outline two possible modes of operations; client/server and Ad Hoc mode WLAN.

An Access Point (AP) is basically a radio base station that covers an area of about 30-300 m depending on the environment. An access point and its associated clients form all together a Basic Service Set (BSS).

In the client/server mode, WLAN terminals communicate with base stations or access points (AP) that form the coverage area. The access points serve the stations in a BSS and they are further connected to the wired network. The set of BSSs together are called Extended Service Set (ESS).

In Ad Hoc mode configuration, the stations communicate directly with each other. In the Ad Hoc mode there is no infrastructure installed and it is easy to operate, but the disadvantage is that the coverage area is limited. Stations in such a configuration are in a Basic Service Set (BSS). Without the ESS the stations operate in an Independent BSS (IBSS).

The 802.11 only specifies the air-interface that is the interface between stations and between stations and Access points. With a distribution system, the coverage area can be extended to whatever depending on the design of the distribution system.

The standard provides the mentioned (client/server, Ad Hoc) services with the following functionality: roaming within an ESS, multiple data rates in BSSs and Power Management (stations can switch off their transceivers to conserve power).

There is no handover mechanism specified in the standard but it introduces a service called re-association, which is related to the roaming from one BSS to another. Two or more adjoining BSS together form an Extended Service Set (ESS) if they share the same ESS identity (ESSID). This is the case when roaming is possible.

WLAN clients establish a radio link with the Access Point that after performing the authentication and authorization procedure will assign the node an IP address giving access to Packet Data Networks (i.e. either private or public networks; Internet). Thus, the node obtains a wireless link and an IP address so it can

communicate with existing IP servers (i.e. DNS) for reaching the services (i.e. web, mail and other application servers) provided in public domain networks.

Chapter 3

Routing in Ad Hoc networks

The IETF defines a set of routing protocols and addressing mechanisms as part of the IP architecture. As described in previous chapters IP is based on building blocks that implement simple functionality that together support a complex architecture.

When considering new technologies such as Ad Hoc networks, new building block are required. The existing routing protocols such as RIP, OSPF, BGP, etc, are not suitable for Ad Hoc networks. The highly dynamic nature of a mobile Ad hoc network results in frequent and unpredictable changes of network topology, adding difficulty and complexity to routing among the mobile nodes. The link state protocols (e.g. OSPF, IS-IS) that work efficiently in fixed IP networks, do not work in Ad Hoc networks. Thus, Ad Hoc networks define a new routing paradigm where the classic algorithms do not work. The IETF has created a working group (i.e. Mobile Ad Hoc Networks MANET [1]) to analyze the Ad Hoc technology and define the required building blocks for continuing patching Ad Hoc networks to the IP architecture. This chapter describes the state of the art in Ad hoc routing.

3.1 Wireless Ad Hoc networks

The wireless Ad Hoc networks can use the WLAN radio technology but in the Ad Hoc mode where the nodes implement a self-organized mesh network. The 3G and WLAN networks are still infrastructure-based networks, thus they utilize the described IETF routing and addressing protocols. However, Ad hoc networks are a

breakthrough technology that allows deploying fully functional networks without infrastructure support. Therefore, new routing and addressing algorithms are required.

The Wireless Ad Hoc networks consist of networks where the nodes move randomly and they create the networks by themselves. These networks will use existing radio technologies such as WLAN (e.g. 802.11a/b/g/e and probably n), 802.15, Bluetooth, etc. When considering the Internet architectures we found that Internet is merely a communications network composed of a collection of systems (nodes), which transfer information between them. Internet differentiates two types of systems; end systems and intermediate systems. The end systems are equipment, which users (either human or machines) may use to access the information at a remote site or equipment. The intermediate systems are computers or network equipment which do not interact directly with users, but forward received data onward towards the intended recipient. One of those intermediate systems is the router. The router consists of a node with one or many network interface cards supporting the IP protocol. The router receives packets from each interface and forwards the received packets to an appropriate output network interface, or to the same network interface but with updated addressing information. The router uses the information held in the network layer header (i.e. IP header; IP destination address) to decide how to forward each received packet. The router forwards packets from one IP network to another IP network but introduces delay (latency) in the packet processing.

In Ad Hoc networks, the end systems take a dual role as intermediate and end systems since each node is acting as a host and a router. Thus, the Ad Hoc networks are a new technology with a lot of challenges since there is no infrastructure and the nodes themselves build the network. The building blocks introduced in IP networks for providing the host their IP addresses (i.e. DHCP), mapping the logical names or phone numbers into IP addresses (i.e. DNS and ENUM) are not available in the Ad Hoc networks. The routing and gateway location protocols (i.e. OSPF, BGP, etc) that are used for building and distributing

the network topology are not suitable for Ad Hoc networks. Thus, all the heavy procedures (e.g. data link protocol frame processing, filtering, selecting the correct output link, queuing delay on input and output links, computing routing tables, etc) required for routing and packet forwarding are now embedded within each of the Ad Hoc nodes.

Therefore, in Ad Hoc networks all problems solved in fixed IP networks such as addressing and naming have to be re-engineered for enabling the networking. Since Ad Hoc networks are on their emerging stage, they are open to re-engineering such as disentangling the idea of identity and addressing. IP addresses are to some extent used for both, which makes device mobility and application mobility more complex. However, separating identity from addressing raises new issues such as how to define the identity Ad Hoc namespace. A non-global namespace of identity IDs with inbuilt mobility capabilities will suit in such infrastructure-less technology such as Ad Hoc networks.

In Ad Hoc networks a new paradigm based on demand routing, where the nodes search for the routes only when needed, suits better than the existing fixed routing mechanism, where the nodes maintain the complete network topology. Security is another problem that already exists in all IP networks but is highly more complex in Ad Hoc networks. The Internet was designed to be “transparent” for carrying packets from anywhere to anywhere without tampering with or modifying the content. This works well when users trust each other or by having intermediate systems that monitor and filter attackers data (i.e. Firewalls, PKI infrastructure [36], etc). The transparency principle used as the basis of Internet does not work in real life.

Thus, in Ad Hoc networks the hosts are also intermediate systems but they may not have the resources for performing the required security functionality. Thus, a new security paradigm based on detecting the misbehavior of certain nodes and neutralizing those nodes is required. Since there is no infrastructure support for

implementing this procedure, the security mechanism should be embedded within the routing protocol and implemented within the cooperative nodes.

Moreover, a security framework for defining the identity at the human or "trusted node" level based on a "bottom-up" identity namespace should be used. Service discovery is another problematic issue that cannot be implemented using existing IP-based protocols that require the fixed infrastructure support or they require having a reliable connection and enough bandwidth. Thus, a service discovery procedure integrated with the routing mechanism is required to locate the necessary services to enable the full connectivity between Ad Hoc and fixed networks.

3.2 Ad Hoc routing

Numerous routing protocols and algorithms have been proposed in the MANET working group, and their performance under various network environments, and traffic conditions has been studied and compared.

Several surveys and comparative analysis of MANET routing protocols have been published [75], [76] and [77]. MANET routing protocols are typically subdivided into two main categories: proactive routing protocols and reactive on-demand routing protocols [75].

Proactive routing protocols are derived from legacy Internet distance-vector and link-state protocols. They attempt to maintain consistent and updated routing information for every pair of network nodes by propagating, proactively, route updates at fixed time intervals. As the routing information is usually maintained in the tables, these protocols are sometimes referred to as Table-Driven protocols.
Reactive on demand routing protocols, on the other hand, establish the route to a destination only when there is a demand for it. The source node through the route discovery process usually initiates the route request. Once a route has been established, it is maintained until either the destination becomes inaccessible (along every path from the source), or until the route is no longer used, or expires.

Most work on routing protocols is being performed in the framework of the IETF MANET working group, where four routing protocols are currently under active development. These include two reactive routing protocols, AODV and DSR, and two proactive routing protocols, OLSR and TBRPF. There has been good progress in studying the protocols' behaviour (almost exclusively by simulation), but the absence of performance data in non-trivial network configurations continues to be a major problem. The perception is that of a large number of competing routing protocols, a lack of WG wide consensus, and few signs of convergence. In order to solve this situation, the intention is to focus the activities of the MANET WG towards the design of IETF MANET standard protocol(s), and move related long-term research work out from IETF. The long-term research work may potentially move to the IRTF organization (Internet Research Task Force) that has recently established a group on "Ad hoc Network Scaling Research" [83].

3.2.1 Proactive routing protocols

The main characteristic of the proactive routing protocols is the constant maintaining of a route by each node to all other network nodes. The route creation and maintenance are performed through both periodic and event-driven (e.g., triggered by links breakages) messages. MANET IETF proactive protocols are: Optimised Link State Routing (OLSR), and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF).

OLSR protocol [78] is an optimisation for MANET of legacy link-state protocols. The key point of the optimisation is the multipoint relay (MPR). Each node identifies (among its neighbors) its MPRs. By flooding a message to its MPRs, a node is guaranteed that the message, when retransmitted by the MPRs, will be received by all its two-hop neighbors. Furthermore, when exchanging link-state routing information, a node lists only the connections to those neighbors that have selected it as MPR, i.e., its Multipoint Relay Selector set. The protocol selects bi-directional links for routing, hence avoiding packet transfer over unidirectional links.

Like OLSR, TBRPF [79] is a link-state routing protocol that employs a different overhead reduction technique. Each node computes a shortest-path tree to all other nodes, but to optimise bandwidth only part of the tree is propagated to the neighbors.

3.2.2 Reactive routing protocols

The reactive routing protocols differ from the legacy Internet approach. To reduce the overhead in legacy Table-Driven protocols, the route between two nodes is discovered only when it is needed. Representative reactive routing protocols include: Dynamic Source Routing (DSR) and Ad hoc On Demand Distance Vector (AODV).

DSR is a loop-free, source based, on demand routing protocol [80], where each node maintains a route cache that contains the source routes learned by the node. The route discovery process is only initiated when a source node does not already have a valid route to the destination in its route cache; entries in the route cache are continually updated as new routes are learned. Source routing is used for packet forwarding.

AODV is a reactive improvement of the DSDV protocol. AODV minimizes the number of route broadcasts by creating routes on-demand [56], as opposed to maintaining a complete list of routes as in the DSDV algorithm. Similar to DSR, route discovery is initiated on-demand, the route request is then forwarded by the source to the neighbours, until either the destination or an intermediate node with a fresh route to the destination, is located.

DSR has a potentially larger control overhead and memory requirements than AODV since each DSR packet must carry full routing path information, whereas in AODV packets only contain the destination address. On the other hand, DSR can utilize both asymmetric and symmetric links during routing, while AODV only works with symmetric links (this is a constraint that may be difficult to satisfy in mobile wireless environments). In addition, nodes in DSR maintain in their cache

multiple routes to a destination, a feature helpful during link failure. In general, both AODV and DSR work well in small to medium size networks with moderate mobility.

3.2.3 Hybrid routing protocols

The hybrid routing protocols integrate the characteristics of proactive and reactive routing protocols and exhibit proactive behaviour given a certain set of circumstances, while exhibiting reactive behaviour given a different set of circumstances. These protocols allow for flexibility based on the characteristics of the network.

The Zone Routing Protocol (ZRP) [55] is a hybrid protocol that integrates both proactive and reactive routing components into a single protocol. Around each node, ZRP defines a zone whose radius is measured in terms of hops. Each node utilizes proactive routing within its zone and reactive routing outside of its zone. Hence, a given node knows the identity of and a route to all nodes within its zone. When the node has data packets for a particular destination, it checks its routing table for a route. If the destination lies within the zone, a route will exist in the route table. Otherwise, if the destination is not within the zone, a search to find a route to that destination is needed.

Despite the large volume of research activities and rapid progress made in the MANET routing protocols in the past few years, this research area still contains many open issues. There has been good progress in studying the protocols' behavior almost exclusively by simulation. Currently, only few measurements studies on real ad hoc test beds can be found [81], [84]. The results from the test beds with real nodes are very important as they are pointing out design problems that were not possible to be detected by simulation (e.g. the so-called communication gray zones problem [82]).

For this reason as part of the research described in this thesis an Ad Hoc framework has been designed and implemented to test the proposed enhancements to enable Ad Hoc connectivity.

Chapter 4

Service Discovery in Ad Hoc networks

This chapter presents the interoperability issues that appear when interconnecting fixed IP and Ad Hoc networks. The interoperability is a well-known problem when connecting different technologies. E.g. the connectivity between Switched Circuit Networks (SCN) and IP based networks is accomplished via signalling and media gateways. However, when considering the interoperability between IP wireless networks with fixed infrastructure and Ad Hoc networks, a new approach has to be considered. This section presents the interoperability between SCN and IP networks as background to analyse the interoperability when considering Ad Hoc and fixed IP networks. As analysed in previous chapter the solution to provide connectivity between Ad Hoc networks and fixed infrastructure consists of defining a service discovery mechanism. This service discovery proposal should work at network layer (e.g. providing information about Gateway and NAT servers, this procedure was called Gateway Location problem at when considering SCN and IP networks). Moreover the service discovery suitable for AD Hoc networks should also provide information at session or application layer (e.g. provide information about DNS, SIP servers that will enable mobility at application layer).

4.1 IP-SCN connectivity

There are some examples of using service discovery (or gateway location) to provide connectivity between different technologies. The interoperability between

IP and SCN networks requires solving several obstacles in terms of addressing and routing. The interoperability between two different technologies is achieved using an intermediate element named “gateway” that bridges both sides [30]. The Gateway can have a simple function of connecting two networks or it can have a more complicated role by performing also a protocol conversion. The gateway functionality is rather complex since it requires making an address lookup and mapping, and even a protocol conversion (IP protocol versus ISUP).

The different address space used in IP and SCN networks [33], [34] is the first obstacle to solve in order to establish connectivity over both technologies. Telephony is the primary service in SCN networks, and it would be the first service affected by this numbering and addressing problem. The telephony service in SCN networks has the counterpart on Internet networks (Voice over IP, VoIP) but the end point uses IP addresses or the textual names (URI) as addressing information.

The end points in the SCN networks are identified using their *directory numbers* and in most cases textual names cannot be entered using a normal telephone, which only has numeric keys (i.e. a black phone). An address entered from an SCN terminal for identifying a destination user must be an E.164 number independently whether the destination terminal addressed originally was identified by an E.164 number (meaning that it can be another device in the SCN network) or it was identified by an IPv4/IPv6 address (meaning that it is a terminal in the IP network).

The terminals attached to IP networks also have to use the E.164 numbers to address a terminal located in the SCN networks. This requires that the E.164 number must be mapped to either an IP-address or a DNS host name to be routed within the IP networks up to the “Gateway” that will perform as the bridge with the SCN networks. A directory will perform this mapping, thus it is the first additional module that will assist the IP protocols on their functioning. The directory (e.g. X.500) consists of a database [28] containing information about the subscribers on one or several networks normally.

The UMTS or 3G networks is an example of IP networks that require connectivity with SCN to enable Telephony sessions with legacy phones. The 3G networks introduced a new set of network entities for interoperating with legacy networks. Those elements are used to perform the routing and message translation within the UMTS architecture towards legacy networks. Those elements are the Media Gateway (MGW), the Media Gateway Control Function (MGCF) and the Roaming Signaling Gateway (R-SGW). The R-SGW updates the information in the HSS when users are roaming between wireless networks (2G, 3G). Border Gateway Control Function (BGCF) is the last element in the 3G networks that acts like a CSCF. The Gateway Mobile Switching Center (G-MSC) has the functions to adapt the signaling from 3G to 2G following the formats used in the Mobile Switching Center (MSC) used in 2G. Home Subscriber Server (HSS) stores user profiles (i.e. similar to HLR and VLR in 2G networks). HSS keeps the user information regarding service profiles, terminal capabilities and so on. Thus, the numbering translation is performed at the HSS. The resulting information will be placed in the SIP URI and will route the call towards the SCN, and the MGCF will be the last entity in the 3G networks that will convert the signaling messages.

When considering interoperability another key issue is to locate the gateway that will perform the addressing mapping and signaling conversion between the connected technologies. Similar problem has to be solved when connecting Ad Hoc networks to the fixed infrastructure. However, Ad Hoc networks cannot rely on fixed servers that enable the connectivity between different technologies. Therefore, a new paradigm has to be defined to solve the gateway location problem that will provide the connectivity between Ad Hoc and fixed IP networks. This thesis proposes considering gateway location [29] as part of the service discovery dilemma in Ad Hoc networks. Extending the semantics of “service”, this proposal considers connectivity as one service that should be provided to Ad Hoc networks to become part of broadband infrastructures. The rest of the chapter presents a novel proposal to implement service discovery in Ad Hoc networks in order to solve the connectivity problem.

4.2 Service discovery in Ad Hoc and fixed IP networks

An intensive research in Ad Hoc networks is taking place. In general, Ad Hoc networks are architectures that can be rapidly deployed without pre-existence of any fixed infrastructure. A critical functionality in these networks is the service distribution. Finding a particular service or resource (e.g. gateway to reach the fixed network, SIP proxy or registrar server, etc) out of thousands of accessible services and devices is already difficult. Thus, to do it in a mobile dynamic environment is challenging. A service distribution mechanism for Ad Hoc networks must find services without relying on a centralized directory server. At the same time, it should minimize message overhead and should be able to deal with the high dynamics and changing topology of these networks [45].

Current solutions on service architectures assume a fairly stable network since they are based on centralized service registries. However, these architectures do not adequately solve all the problems that may arise in a dynamic domain, since frequent disconnections inherent in Ad Hoc networks may lead to inconsistency of data in centralized service directories. Available solutions such as JINI [46], UPnP [47], Salutation [48], OSGi [49], and SLP [50] among others focus on application layer services without considering connectivity as an essential service since it is granted by default in fixed networks. Moreover, these existing proposals are based on central directories. The existing literature offers other architectures for service distribution in Ad Hoc networks. Some solutions are based on mesh and tree structures built from multicasting. This thesis presents a proposal that is not based on central directories or agents, and differs from other architectures in the mechanisms used to build the service backbone. The interoperability of Ad Hoc networks require the creation of a backbone infrastructure that will allow Ad Hoc nodes to connect to the fixed IP networks. The proposed mechanism enables the attachment to fixed IP networks and extends existing services (e.g. Telephony, Web Services, File Transfer, Messaging, etc) over Ad Hoc networks.

4.3 Service discovery for Ad Hoc networks

In the proposed mechanism, any node within the network can act as a service provisioning node or a server, which only requires a registry for storing the service description and the mechanism for the service discovery. The discovery technique enables services to announce their capabilities and allows the clients to find and use the needed services.

The service distribution is based on a set of core nodes, which we refer to as the backbone. The backbone constitutes the infrastructure to provide service discovery information, thus relieving non-backbone nodes from participation in this activity.

Ad Hoc networks require a higher degree of freedom and a fully decentralized architecture that is provided by a service backbone. The complexity of the Ad Hoc networking can be hidden behind the proposed service backbone, which provides transparent access to both local and remote services in a uniform way. The service registry implemented by the backbone should immediately reflect changes affecting service availability, and services that are no longer reachable, should not be available for discovery.

The proposed architecture for service discovery is based on the creation of a service backbone that is formed using two different methods (i.e. ***Hierarchical*** and ***Distributed*** backbone). The Hierarchical method differs from the Distributed one in the algorithm for organizing the nodes when creating the service backbone.

4.3.1 Hierarchical backbone

Flooding a service request in an ad hoc network can be very costly in terms of network throughput efficiency, as well as node energy consumption. Other solutions based on backbones at the network layer require multicast capabilities, which are not always available. Thus, the preferred approach is to build the service distribution mechanisms on top of the network layer in order to take advantage of the topological structures defined at that layer. For example, if multicast protocols exist, then the backbone defined at the network layer is re-used for service

distribution. If it is not the case, then we propose to organize the service distribution architecture into a hierarchical structure built with an efficient clustering algorithm. This approach strengthens interoperability with different routing algorithms.

4.3.1.1 Service Backbone Creation

When no other strategy is defined, local groups or clusters of mobile nodes will be formed where one node will be elected dynamically to become the cluster head in each group. The set of elected cluster heads will form the service distribution backbone, where the function of each cluster head will be to track available services within its local group. Current algorithms for the construction of clusters are contained in many routing protocols such as CBRP [69] and GSR[70]. Also other clustering heuristics, such as the highest-degree [71] and the Linked-Cluster Algorithm (LCA) [72], have proactive strategies. By proactive, we mean that they require a constant refresh rate of cluster dependent information.

Therefore, for the hierarchical backbone creation the use of a non-periodic procedure for cluster-head election such as the Weighted Clustering Algorithm (WCA) [73] is proposed. This algorithm not only is invoked on demand by the nodes with higher mobility, but it takes into consideration the ideal number of nodes that a cluster can handle, the mobility (speed of nodes) and the battery power, in order to assign weights that will be used to determine the cluster head nodes.

4.3.1.2 Service distribution

In this model, when a client node needs a service, it browses its service cache memory, and if the service is not found, it sends a service request to its cluster head. The cluster head verifies if the service is available in its group to reply to the client node. If the service does not exist in the cluster, then the cluster head re-sends the service request to other available cluster head nodes until the service is found; in which case, the client node receives a reply-message from the cluster head where the service is available. This message will contain the necessary

information, such that the client node will establish a peer-to-peer communication link with an appropriate routing protocol. When a service request is sent, a timer is triggered to limit the time the client node waits for a reply-message. If the timer expires and no reply was received, it can re-send the service request until a maximum number of times (the timer may be increased in subsequent requests). At this moment, if no reply-message is received, the client node assumes the requested service is not available.

When a service becomes available, the server node advertises it to its neighbors by using a distance-limited flooding mechanism (using a TTL=n, where n represents the maximum number of hops that a packet can traverse). The server node periodically announces its services to its designated cluster head, so it can keep track of the services in its group. If the cluster head does not receive these updating messages in a period of time, it assumes that the service is no more available and erases it from its registry.

4.3.1.3 Service access

In the Hierarchical strategy, once the service is found using the service backbone, the application will figure out the service invocation mechanism to use that service. The application will indicate the service access models to be used, such as Jini RMI (Remote Method Invocation) [46], UPnP, or any other communication middleware that allows heterogeneous applications to communicate.

4.3.1.4 Simulation analysis

In order to evaluate this first proposed architecture, the necessary software extensions in GloMoSim [74] were developed. Two different strategies are considered for service distribution: *flooding* and *clustering* (either with LCA or WCA). For a simple evaluation we defined the scenario as follows. The node density was increased from 10 to 50 nodes in a simulation area of 500 x 500 meters; the transmission range was set up to 250 m, 802.11 was used as the MAC protocol and AODV [56] as the routing protocol. For mobility, Random Way Point Model (RWPM) with pause time of 0 seconds and 10 m/s as maximum speed

was used. The evaluation is repeated with different initialization values until the variance of the results obtained was less than 7%. Figure 13 shows that in this scenario, clustering presents a better behavior than flooding; but even more, not every clustering algorithm presents the same performance as node density increases. The pro-activity of some clustering procedures such as LCA (and other algorithms from hierarchical routing protocols) results in a worse behavior even than flooding as node density increases.

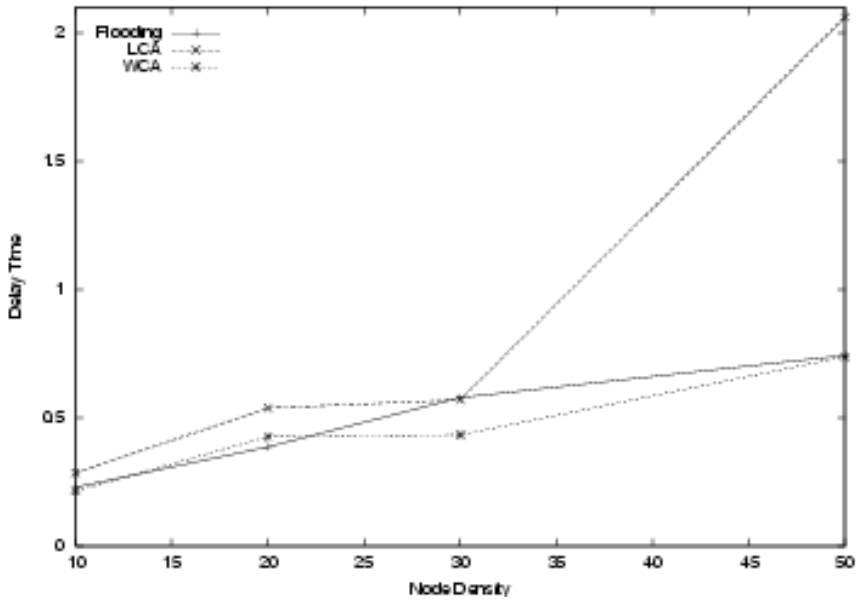

Figure 13. Delay on service distribution.

Figure 14 shows a comparison of the overhead generated by the two clustering algorithms.

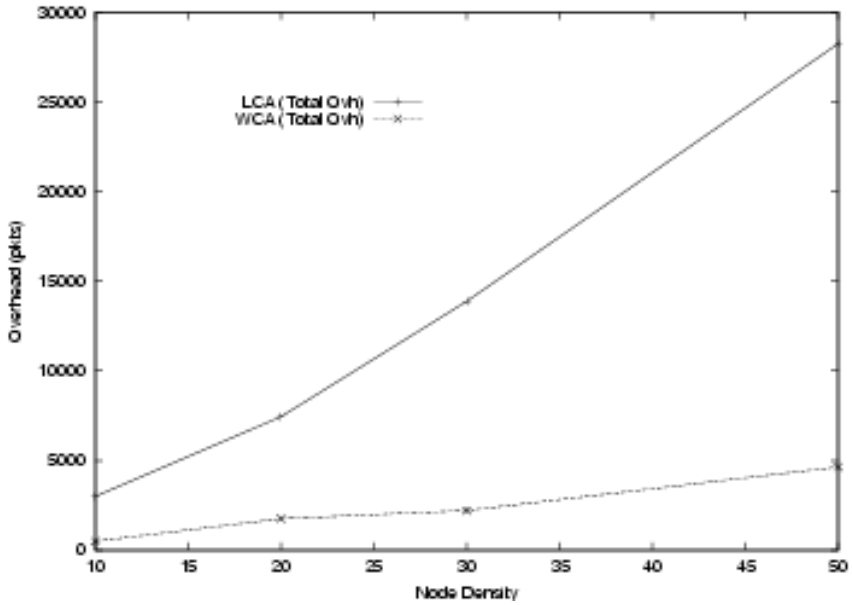

Figure 14. Service distribution overhead.

4.3.2 Distributed backbone

In this thesis we propose a distributed backbone model differing from the Hierarchical backbone in the algorithm for creating the service distribution infrastructure. This approach is based on the concept of node taxonomy [45] that defines two types of nodes [51], [52]; "smart" and "dummy" nodes. This node taxonomy that differentiates the nodes that are capable of being part of the service backbone, from the nodes that will use it.

This architecture does not impose any strategy for organizing the nodes that form the service backbone. The proposed fully distributed service backbone only imposes a service backbone access algorithm based on the network conditions. This architecture is based on the node taxonomy and the willingness of each node to contribute to the service distribution. The node taxonomy differentiates the nodes that are willing to implement the service backbone (i.e. "smart" nodes) from the nodes that only access the services using the service backbone (i.e. dummy nodes). When entering the network the nodes decide by themselves whether they can become "smart" nodes or they remain as "dummy" nodes, which is the default state. "Smart" are nodes with enough resources, running multiple protocols

simultaneously and willingness to maintain route and service information. “Dummy” nodes are devices with limited resources, running a single Ad Hoc MANET protocol and having a single network interface.

The “smart” nodes compose the backbone and implement a link state routing protocol between them for sharing network state information. The rest of nodes accessing the service discovery mechanism will be “dummy” nodes. All nodes can be equally part of this “service backbone” and they can attempt to become “smart” node at any time. The “smart” node will issue a broadcast message (TTL=1) to perform a network attach, which consists of discovering other “smart” nodes present in the network and join the service backbone. Moreover, in this broadcast message, the “smart” node is also announcing its services to all the nodes in the network. However, the existing “smart” nodes will apply a service backbone access algorithm, which depends on the network conditions. Therefore, “dummy” nodes that normally act as service clients can become “smart” nodes and if accepted they can be included into the “Ad Hoc service distribution backbone”. The “smart” nodes resign from the service backbone and can fall back to become “dummy” nodes at any time when their resources get exhausted or they terminate the service provisioning.

4.3.2.1 Service backbone creation

The “Ad Hoc service distribution backbone” is formed with one or many “smart” nodes. The requirement for being a “smart” node is to have enough resources and the willingness for contributing to the service backbone. Moreover, in order to be a “smart” node it is required to implement a novel “multiprotocol” architecture. The proposed “multiprotocol” architecture is necessary to perform a link state routing protocol between the “smart” nodes but still communicate with “reactive” or “on demand” routing protocol with the “dummy” nodes. This architecture benefits from having multiple routing algorithms running simultaneously in the node.

The most important feature is that this architecture based on the "smart" and "dummy" node taxonomy does not add any requirement to the nodes with limited resources ("dummy"). The "dummy" nodes may be running a single IETF MANET protocol and do not notice the existence of "smart" nodes that are helping to the overall network performance. Instead all the "service backbone" creation is assigned to the nodes with enough resources ("smart") that will assist the overall routing and service location process in the network. The "smart" nodes will interact with the native IETF MANET nodes and will cooperate with them in order to extend the network lifetime.

This fully distributed architecture only requires having a certain amount of "smart" nodes for implementing the service backbone. However, the "service backbone" creation applies an access restriction algorithm to the "smart" nodes that intend to join the "service backbone".

Recent publications claim that the maximum number of simultaneously successful transmissions is in the order of the number of nodes (i.e. number of nodes randomly distributed in the Ad Hoc network $0 \leq N \leq \infty$.). It was proven [53] that the number of simultaneous successful transmissions has an upper bound (i.e. Number successful transmission: Nt). This upper bound is dependent of the processing gain of the Ad Hoc network (i.e. *G*) and the reception signal Interference noise ratio (SINR) threshold (i.e. *b*). The processing gain, in practice G, considers the processing gain on each node including the gain from the antennas when receiving and sending the messages. The gain is influenced by the capabilities of the nodes to process the message and the available resources to handle the messages efficiently.

Eq. 1: $Nt = O\left(\frac{G}{b}\right)$

The "smart" nodes are nodes with more resources, which means that they increase the overall gain in the network. Example: In real networks the possibility of having few "smart" nodes randomly distributed in the Ad Hoc network will increase the

number of successful transmissions (i.e. Wireless Access Points attached to fixed infrastructure and with enough capacity and resources).

Nonetheless, the “smart” nodes communicate between them using link state messages increasing the average noise signal in the network. There is a trade off in the number of “smart” nodes active in the network in order to increase the number of successful transmissions in the network.

Therefore, a “service backbone” access algorithm is required for obtaining an optimal throughput in the network. Thus, having few *smart* nodes that keep high degree on capacity and gain and low signal noise ratio in the network will allow an Ad Hoc network to have maximum number of simultaneous successful transmissions. An access mechanism to limit the number of *smart* nodes in the backbone is required to avoid having an excessive number of *smart* nodes. Thus, when the number of nodes that form the backbone reach an optimal number based on Eq.1 maximization, the new smart nodes that after joining the Ad Hoc network intend to be part of the backbone, should be rejected.

4.3.2.2 Service distribution

The “smart” nodes that implement the service backbone maintain the service information between them using link state protocols (e. g. OLSR [54]). Moreover, the “smart” nodes are organized in clusters. These clusters will use a link state mechanism between them for sharing the service information. The services clusters will use a reactive mechanism for discovering new services in other clusters. Thus if the local cluster head has received a service request that is not available in its own cluster, the cluster head will issue a reactive request to the neighbor clusters asking for the service (i.e. ZRP [55]). If the service is not found using this algorithm a service error response (i.e. Service Unavailable) will be returned to the client that was requesting the service.

4.3.2.3 Service access

Any node (i.e. "dummy") acting as service client can issue a service request. The nodes will not notice the existence of "smart" nodes except in the overall functioning of the Ad Hoc network. The nodes will utilize higher layer service discovery mechanisms (e.g. JINI, UPnP, etc) that will be translated into a lower layer service discovery request (e.g. the JINI operation for discovering a service will be mapped into "*ServiceDiscovery_Request"* like in reactive protocols such as AODV [56]). A "smart" node will respond to the message query providing the address of the node provisioning the requested service. This service discovery request is broadcast (TTL=1) to all the neighbors. Any intermediate "dummy" node that receives a service query and has the address of a "smart" node in its cache, will unicast the service query to that specific "smart" node that will attend the query. Once a service is found or the node that provides the service is located, the service invocation mechanism to use that service is provided by any other communication middleware that allows heterogeneous applications to communicate (i.e. JINI, UPnP, etc).

4.3.3 Discovery procedures

The proposed distributed service backbone requires a node taxonomy composed by "smart" and "dummy" nodes. This section presents the algorithm that each node has to perform for implementing service discovery and service provisioning.

"Smart" node:

- **Network attach procedure** is executed every time a "dummy" node becomes a "smart" node. The "smart" node sends a broadcast with TTL=1 using the local underlying routing mechanisms for implementing the service discovery procedure. In this message the candidate "smart" node is announcing its capabilities and available services. If the candidate "smart" node does not receive any response it means that no other "smart" node is reachable in the same region either directly or through any local "dummy" node. In case the "smart" node receives a response from an existing "smart"

node, a service backbone creation is initiated. The service backbone creation consists of setting a link state relationship among the smart nodes for the service distribution in the local region and a clustering infrastructure with other “smart” nodes for the service discovery in neighbor regions. The responding “smart” node will indicate the new node whether it can join the “service backbone” or not depending on the access algorithm (i.e. depending on the number of nodes in the network and the number of nodes that already joined the “service backbone”).

- **Network attach reception** occurs when a “smart” node receives a request from a candidate “smart” node, which is performing the network attachment. The request is received either directly from another “smart” node or indirectly via an intermediate “dummy” node. The “smart” node caches the address of the new candidate “smart” node in the network. The “smart” node executes the “service backbone” access algorithm and sends back a unicast message to the originating candidate “smart” node informing whether it can become a “smart” node and join the “service backbone” or not. If the new node can join the “service backbone” a link state relationship between the nodes is established.

- **Service discovery reception** occurs when a “smart” node receives a service request directly from a “dummy” node or from another “smart” node.

  a) The “smart” node may receive the service request in a reactive request from a “dummy” node. The “smart” node will contain all the service information about the local cluster formed by all the “smart” nodes located in the region nearby. If the “smart” node does contain the address of the server that provides the requested service, it will insert it in the service response sent back to the “dummy” node. If the contacted “smart” node contains the requested service it will be indicated directly via unicast to the “dummy” node that issued the service request. If the “smart” node does not contain the address of the requested service provider, the service request will be forwarded in a reactive manner to the neighbor service clusters. The “smart” node times out if a service response from the neighbors cluster is not

received within a certain period of time. The “smart” node will respond to the “dummy” node with service error or service unavailable message.

b) If the “smart” node receives the service request from another “smart” node it means that the neighbor cluster does not contain the service and it is implementing a reactive service discovery over all the adjacent service clusters. If the contacted “smart” node contains the requested service it will be indicated directly via unicast to the “dummy” node that issued the service request. If the “smart” node does not contain the address of the server, the service request will be forwarded according to the multicast tree formed by the neighbor cluster heads (e.g. ZRP mechanism may be utilized for implementing the service clustering).

“Dummy” node:

- **Service discovery** process is executed when the node needs to find out the server that is provisioning a certain service. The service discovery process will be implemented using the specific service discovery mechanism provided by the routing layer.

- **Network attach reception** occurs when a “dummy” node receives the broadcast message sent by a “smart” node entering or attaching to the network.

  a) The “dummy” receives the message and caches the “smart” node address as gateway or service provisioning server.

  b) If the “dummy” node already has the address of another “smart” node it will be returned to the “smart” node that sent the broadcast.

- **Service discovery reception** occurs when a “dummy” node is sending a service request broadcast because it did not have the address of any “smart” node.

  a) If the contacted “dummy” node has the address of any “smart” node, the service request will be forwarded to that “smart” node that will respond

directly to the “dummy” node that issued the request with the address of the server provisioning the service and service invocation mechanism (e.g. JINI, UPnP, etc). This node should not wait to see if there was an answer to the request and it has to forward the service request immediately to the smart node in order to provide robustness to the service query due to dynamic nature of Ad Hoc networks. The node that issued the request will dismiss the response if they contain duplicated information.

b) If the contacted “dummy” node does not have the address of any “smart” node, the service requested is broadcast following the reactive service discovery mechanism.

The proposed *distributed backbone* is tightly related to the node taxonomy and needs the existence of “smart” nodes willing to provide the service layer, and nodes that require service provisioning. Thus, the “smart” nodes should have certain motivation for participating in the creation of the “service backbone” and thus contribute to the Ad Hoc network (e.g. service provisioning with charging requirements, operator support for accessing the core network services (UMTS), etc.). This method provides a non-reliable “service backbone” based on a best-effort approach depending on the contribution from as many “smart” nodes as possible within the optimal limit defined by Eq. 1.

Thus, the service provisioning is moved to a different layer (i.e. application layer or middleware provisioning) where depending on the interest from certain nodes the “service infrastructure” will be automatically built or not. This approach requires a cross layer architecture where the service discovery is linked with the routing layer. Thus, all nodes would have a service layer where the service discovery will trigger the specific routing functionality built for this purpose (i.e. AODV service extension or service information in link state protocols). The main point of this proposal is that nodes with low resources (i.e. dummy) do benefit from the service layer without significant additional requirements to their normal behavior.

The proposed node taxonomy defines two logical entities (i.e. dummy and smart nodes) that facilitate the service discovery and connectivity between Ad Hoc and

fixed networks. The smart nodes will act as the service providers or service brokers (e.g. help to find the service providers) when considering service discovery in Ad Hoc networks. When considering connectivity, the smart nodes will act as Gateways either to reach nodes in the fixed networks from the Ad Hoc nodes or to reach Ad Hoc nodes from nodes in fixed networks. The smart nodes acting as Gateways to the fixed infrastructure have additional requirements in order to provide connectivity to the Ad Hoc nodes. The proposed Gateway will act as NAT collocated with a DHCP for providing global addressing and translation into the private addressing space utilized within the Ad Hoc networks. Moreover, in order to make the Ad Hoc nodes visible to the fixed infrastructures the Gateway that provides the connectivity has to publish its address in the BGP tables.

This thesis defines the mechanism for creating a service provisioning backbone based on novel node taxonomy. The functionality of the smart nodes can vary from simple service provider to more complex connectivity gateway, but this is out of the scope of this thesis.

Chapter 5

Ad Hoc Framework Implementation

In previous sections a novel mechanism to implement service discovery in Ad Hoc networks to extend the connectivity towards fixed IP networks has been proposed. As described in previous sections, experimenting existing (and novel) routing protocols in test beds is very important to deeply understand their behavior in a real environment. Existing test beds [84] have demonstrated un-expected behavior of routing protocols that were not identified with simulations. “Gray zones” [82] and route instabilities are detected only with implementations on real nodes since the simulations do follow predefined mobility models and stable radio propagation. The protocol scalability, sensitiveness to users’ mobility patterns and speeds are difficult to investigate on simulations. A simulation permits the study of system behavior by varying all its parameters, and considering a large spectrum of network scenarios. Therefore, after preliminary design of the service discovery proposal, a tested implementation is required in order to validate the proposal. In addition to the basic routing framework a widely used application in fixed networks such as telephony is required to analyse the behaviour of the design in real environments. This chapter presents the Ad Hoc framework design to be used as the tested for validation and performance analysis.

5.1 System evaluation

Evaluating a system performance with a model consists of two steps:

To define the system model, and to solve the model using analytical and/or simulative techniques.

Analytical methods are often not detailed enough for the ad hoc networks evaluation and in terms of accounting for mobility, they are in their infancy. A simulation modeling is a more standardized, mature, and flexible tool that allows (by running the simulations) data collection to analyze the protocol performance in most cases. A very large number of simulation models have been developed to study ad hoc network architectures and protocols under many network scenarios (number of nodes, mobility rates, etc.). The existing simulation studies compare and contrast large number of routing protocols in a theoretical manner. The simulations outperform the real behavior of the protocols and important divergences between simulators (e.g. NS-2 and Glomosim) running the same protocols with similar conditions were found. The observed differences are not only quantitative (different absolute value), but also qualitative (different general behavior) making the validity of some of the existing simulation studies questionable (Figure 15 and Figure 16) [87].

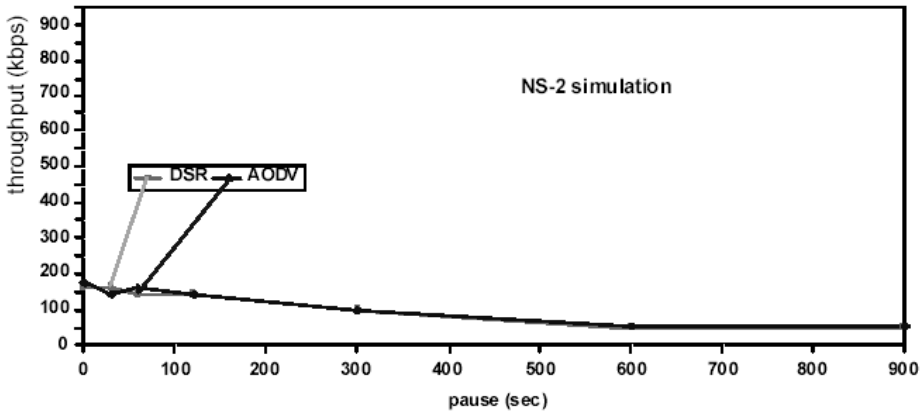

Figure 15. DSR and AODV protocol simulation with NS-2.

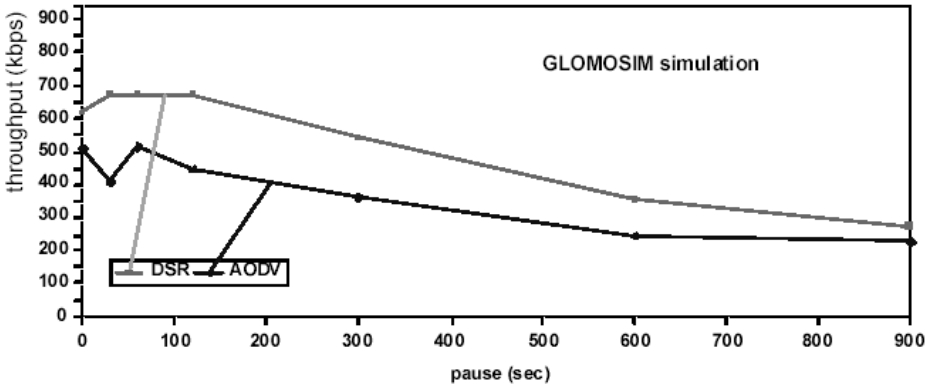

Figure 16. DSR and AODV protocol simulation with Glomosim.

5.2 Ad Hoc framework

After defining the service discovery algorithm a software package implementing an Ad Hoc framework was used as a test bed for validating the proposal. The main goal of this software framework is to develop a test bed for the proposed service discovery backbone in order to provide interoperability with fixed IP networks. The state of the art in terms of similar software packages implementing Ad Hoc frameworks and test beds, contain a routing module that includes a single routing protocol. This routing module may add new routing protocols but still there is a single protocol executed at any time [81]. Therefore, the proposed software package that implements the Ad Hoc framework is based on a novel approach. The Ad Hoc framework software design follows a "Data Replication" architecture model, where there is a centralized server that maintains the data provided by multiple clients or actors. The server is the routing cache and the clients or actors are the software components that implement the routing protocols. The data is processed according to the input received by the clients and the resulting data is accessible to other clients to execute certain algorithms. Figure 17 shows the Data replication software architecture model.

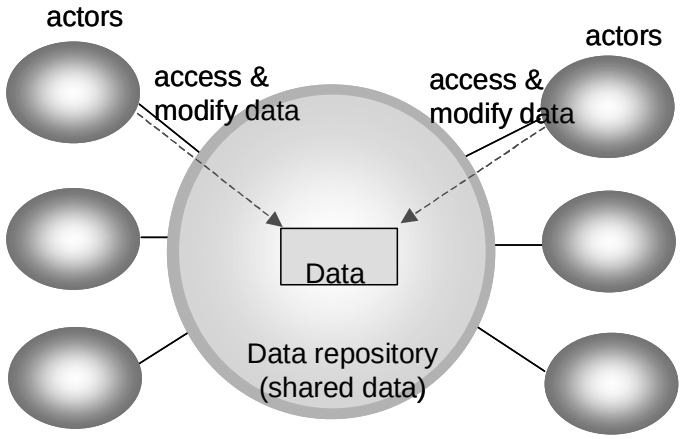

Figure 17. Data replication SW architecture model.

In order to test the service discovery based on the node taxonomy a novel "multiprotocol" architecture to implement "smart" and "dummy" nodes is required. The node taxonomy differentiates nodes with low resources running a single IETF MANET protocol, versus "multiprotocol" nodes with enough resources that assist

the overall routing process in the network. The “smart” “multiprotocol” nodes will interact with those native IETF MANET “dummy” nodes and will cooperate with them in order to extend the network lifetime.

In the “Data Replication” architecture, the routing cache corresponds to the centralized server that maintains the routing data. The multiple protocols running in the node correspond to the different clients that can access the data in the cache.

The Ad Hoc routing framework consists of a software package, which can support different Ad Hoc network routing protocols, such as proactive, reactive and also some hybrid solutions. The Ad Hoc routing framework can be installed in a node (PDA or Laptop) that runs the Linux Operating System. With this framework, we could add new routing protocols and other functionalities, such as naming and service discovery. This section describes the components of the Ad Hoc framework and all the subsystems including interfaces and the basic functionalities implemented in the framework. The framework provides general functionalities for both proactive and reactive routing protocols. In addition to those general functionalities, a reactive protocol (e.g. AODV [56]) and a proactive protocol (e.g. OLSR [54]) are implemented on top of the framework. The framework implementation is integrated into a small number of Personal Digital Assistant (PDA) nodes (iPAQ) for testing and validation.

The Ad Hoc Framework comprises four subsystems. The Common module includes all the modules that must be kept constantly running in the node since they are used by the modules that implement the routing protocol. The Reactive Routing subsystem that includes all the modules required for implementing reactive routing protocols (e.g. AODV). The proactive routing subsystem includes all the modules that implement proactive routing protocols (e.g. OSRL). Finally, the service modules include all modules required for implementing the service discovery mechanism at the routing layer. At this stage the Ad Hoc framework implementation includes the following modules within those four subsystems. The

four subsystems are Ad Hoc framework categories while the modules within those subsystems are concrete software components as depicted in Figure 18.

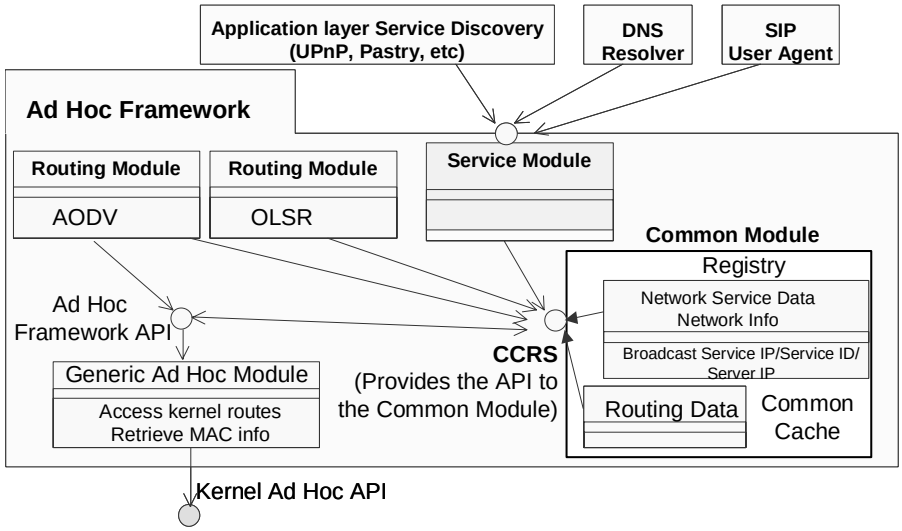

Figure 18. Ad Hoc framework architecture.

The Common modules are Common Cache Registry Server, Common Cache and Registry. The reactive routing modules include the ODRM module and the AODV module. The proactive routing modules include the OLSR module. The service module is not integrated yet.

5.2.1 Common Modules

The common modules are equivalent to the centralized "Data Repository" and consist of a Registry, the Common Cache and the Common Cache Registry Server that communicates with the independent routing modules. The Common Cache collects routing and other information from the routing protocols running in the node. The Registry stores information of the routing protocols running in the node. The Ad Hoc framework has been designed in order to contain several independent routing protocols running simultaneously as daemons (at this point the framework contains two routing daemons: AODV and OLSR). The Common Cache and Registry Server (CCRS) act as the front end of two data repositories: the Common Cache and the Registry. Thus, the routing protocols running daemons as clients

may access the Common Cache and the Registry via communication with the Common Cache and Registry Server.

The Common cache stores all the route information that the CCRS receives from all routing protocols running in the node as daemons. The Registry contains the state information of all routing protocol daemons. The state information consists of configuration parameters of the protocol that is active but also other parameters for sending messages to the daemon in order to change its configuration during runtime.

5.2.2 Common Cache Registry Server

The Common Cache Registry Server (CCRS) is one of the most important common modules in the Ad Hoc framework. The CCRS keeps listening to messages coming from the routing protocol daemons that want to communicate either with the Common Cache or with the Registry.

Thus, when one routing protocol daemon starts running, it must upload its state information and other protocol parameters into the registry through CCRS. Moreover, during the lifetime of the routing protocol daemon when the protocol updates its own route table, it must also update the common cache through the CCRS. Thus, the Common Cache always keeps the routes discovered by the multiple protocols running in the device. Meanwhile, the registry keeps the latest state information of the routing protocol daemons running in the node.

When one routing daemon wants to know the status of other daemons, it can ask the information from CCRS. Different daemons can also communicate with each other through CCRS. This could add communication overhead between daemons. But if there is not much communication between daemons, this makes the design neat and clear. The communication between the daemons and CCRS is implemented by different messages defined in request/reply format. The CCRS server and clients are running in different processes and they communicate through a well-known port. The actual state of the Ad Hoc framework implementation contains the following components.

5.2.3 Common Cache

The Common Cache is a hash table that keeps routing and other node related information such as FQDN, services available in that node, etc. The hash table consists of a collection of key-value pairs. The key of each pair is something uniquely associated with the corresponding value. For our common cache this key is the IP address because they are uniquely associated with a node.

5.2.4 Registry

The Registry is a static file that stores information about the routing modules running or that had been running in the node.

The common registry is a component of the common Ad Hoc module within the Ad Hoc framework that will store information on several protocols in the same node. This registry stores specific information from each protocol and common information as last running time, status of the protocol, most efficient configuration information of protocol running last time, etc.

5.2.5 Reactive routing modules

The Ad Hoc framework may contain multiple routing protocols running simultaneously. Thus, the Ad Hoc framework package differentiates the subsystem that will contain reactive modules from the subsystems that will contain the proactive or hybrid modules. The reactive routing modules subsystem includes the software components that provide the interfaces and services required for implementing reactive routing protocols. The reactive routing modules require direct interaction with the Operating System running in the node. Most operating systems (i.e. UNIX/Windows/Linux) have a build-in network stack that resides in the kernel space of the operating system. The routing function is based on the network stack and also resides in the kernel space. Thus, the Ad Hoc framework implementation requires access to the operating system kernel and even some modifications.

AODV module

This section describes the module that implements the specific AODV routing algorithm [65]. We select AODV because it is the most widely implemented reactive algorithm and might become IETF standard.

The AODV module is based on AODV-UU [58]. The actual implementation has been improved and is stable without major bugs. The AODV is implemented as an independent routing module that only provides the AODV protocol logic.

5.2.6 Proactive routing modules

The proactive routing modules consist of the list of independent routing modules that implement proactive Ad Hoc protocols such as OLSR, Fisheye [88], etc. These protocols can be implemented as standalone modules and then integrated into the Ad Hoc framework.

OLSR

OLSR is a proactive routing protocol [54], which means that the protocol periodically sends control messages to maintain the knowledge of the network topology. OLSR protocol is a ``link state" protocol; this means a node broadcasts over the network the list of its neighbors. In this case all the nodes know the neighborhood of all the nodes. Therefore, the nodes have all the routes and thus the shortest path to all the destinations. The OLSR protocol is an optimization of a pure link state protocol. This optimization is based on two premises. First, it reduces the size of control packets: instead of all links, it declares only a subset of links with its neighbors who are named as *Multipoint Relay Selectors*. Secondly, it minimizes flooding of this control traffic by using only selected nodes, *Multipoint Relays* to diffuse its messages in the network.

The OLSR module implements the OLSR protocol as part of the Ad Hoc framework [66] and contains several modules. Figure 19 shows the module diagrams of the system. It also shows the interdependency between modules and where external framework modules are needed.

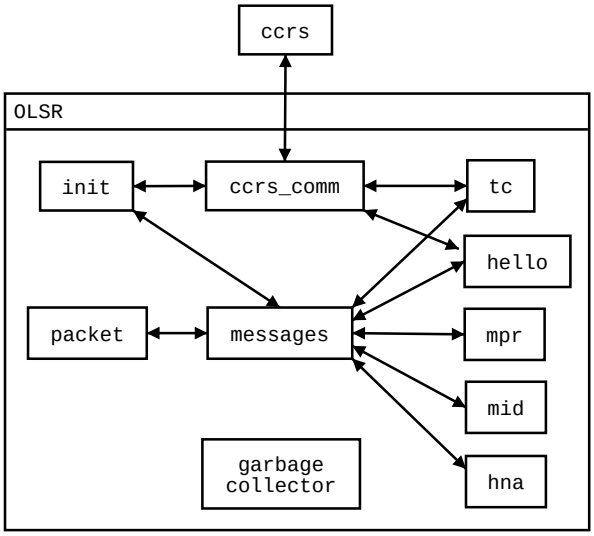

Figure 19. Module diagram of the OLSR protocol.

5.2.7 Service module

This module contains the mechanism for inserting the service information into the routing packets. Thus, embedded with the routing, the node will distribute the service information together with routing information. Figure 20 shows an example where SIP proxy information can be inserted in the service extension.

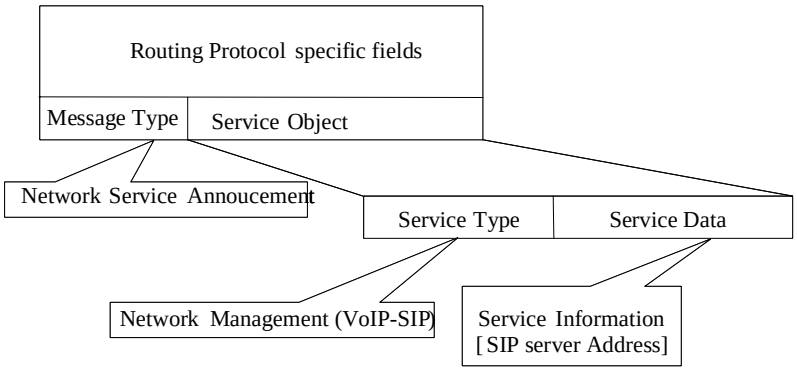

Figure 20. Data structure extension to enable Network Service Discovery at routing layer.

5.2.8 Results and conclusions

The test results from the preliminary implementation confirm the reasoning discussed as motivation for the nodes taxonomy and distributed service backbone. The idea of having “smart” and “dummy” nodes is also supported by economic

factors. The “dummy” nodes will be cheaper than the “smart” nodes and presumable will be part of the mass market, while the “smart” nodes will more expensive and will be implemented by operators or other service providers. Figure 21 shows the results of the test where all the nodes are running OLSR. A second test is performed with all the nodes running AODV. Finally, the third test includes a single “smart” node while the rest of nodes are “dummies” with lower resources (i.e. 6 Ad Hoc nodes; 5 iPAQs and 1 laptop). The packet loss in the test with all the nodes running only OLSR is higher than the test where all the nodes are running AODV. In high mobility conditions bigger packet loss occurs in case of OLSR because of the latency during the link state update. So, if the link is broken a packet will be lost until the link information is updated in the routing table. The test shows that after introducing only one “smart” node the results are favorable. However, AODV running alone has the lowest packet since the route is reactively discovered and is fresh enough so that no or very low packet loss will occur. Introducing only one “smart” node (i.e. running AODV and OLSR) in the network increases the overall packet loss even some of the nodes are running OLSR and others AODV.

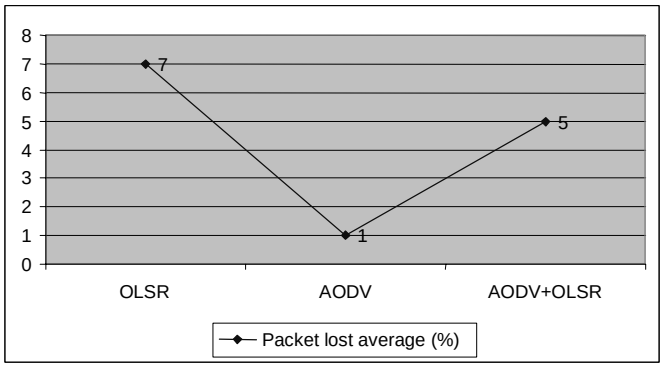

Figure 21. Packet loss in test bed.

Figure 22, shows the results of the test in terms of delay when obtaining new routes or delay when sending a service request. The delay in obtaining the new route or service information is higher in AODV than in OLSR. A reactive protocol has to issue every time a new request that has to be broadcast and attended by any intermediate recipient until the request reaches the intended destination that will send back the response. In proactive protocols that information is already available in the cache so the delay is much lower.

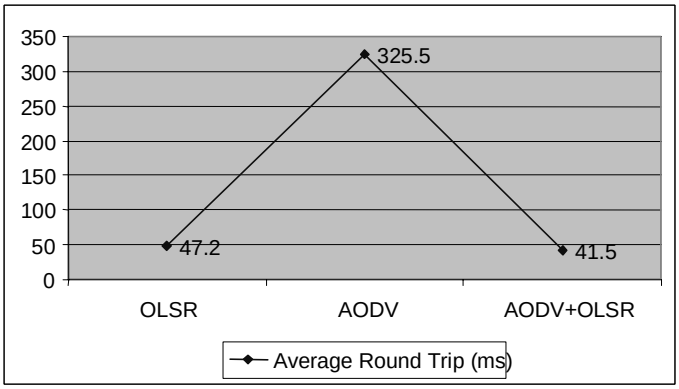

Figure 22. Data packets round trip.

Figure 22 shows that by including a single "smart" node in the network, the delay decreases considerably. The "smart" node attends the route or service request so the request does not need to cross the entire network before the request reaches the intended recipient.

Chapter 6

Application layer connectivity

The proposed mechanism to extend Ad Hoc networks connectivity at link layer has been presented in previous chapters. An Ad Hoc framework has been implemented to demonstrate the concept in real environments. This chapter presents the mechanism for providing connectivity at application layer using the Session Initiation Protocol (SIP). This chapter provides the results of testing the Ad Hoc framework together with SIP protocol to implement a leading application with demanding requirements in terms or routing such as IP Telephony.

6.1 Connectivity at application layer

Previous chapters focused in the connectivity at IP layer where the proposed service discovery allows finding the right nodes that enable the routing across different network technologies. The application layer connectivity consists of providing visibility to nodes with dynamic IP address by using application layer naming (e.g. URI).

6.1.1 Naming in IP networks

The textual names that replace the IP addresses have different formats depending on whether they are referring to a host or another resource in the network. The IETF has defined the syntax for identifying the end points, hosts or resources over the Internet. The Uniform Resource Identifier (URI) [13] is a string of characters that are used for naming or identification of those hosts or any abstract or physical resource.

The semantics of the URI are derived from the concepts introduced by the World Wide (WWW) for identifying resources over the Internet [15].

The main advantage of using textual strings like URI is usability. The URI allows the introduction of new types of resource identifiers that can be reused in many different contexts.

The URI is used in IP networks for identifying a resource that can be anything that has an identity (i.e. an electronic document, an image, a service, a human being, a host, a machine, etc). The URI consists of an object (string of characters) that is used as reference or identifier of the resource. Thus, the resource is conceptually mapped to an identifier, and the resource can remain constant even when the content changes.

The URI is written as follows: <scheme>://<authority><path>?<query>

There are many different methods for accessing the resources and the scheme defines the semantics for the interpretation of the remainder of the URI string.

Examples:

ftp://ftp.is.co.za/rfc/rfc1808.txt

gopher://spinaltap.micro.umn.edu/00/Weather/California/Los%20Angeles

http://www.math.uio.no/faq/compression-faq/part1.html

mailto:mduerst@ifi.unizh.ch

sip: pep@netlab.hut.fi

tel: +358405201815

However, there are many variations and some components may be absent (i.e. some URI schemes do not allow an <authority> component, and others do not use a <query> component).

A generic syntax for the “absolute URI” is as follows:

absoluteURI = scheme ":" (hier_part | opaque_part)

The URIs that are hierarchical in nature use the slash "/" character for separating hierarchical components.

hier_part = (net_path | abs_path) ["?" query]

net_path = "//" authority [abs_path]

abs_path = "/" path_segments

The URI schemes include a top hierarchical element for a naming authority. The authority administers the namespace defined by the remainder of the URI.

authority = server | reg_name

The authority component is preceded by a double slash "//" and is terminated by the next slash "/", question-mark "?", or by the end of the URI. Within the authority component, the characters ";", ":","@", "?", and "/" are reserved.

There are other URI schemes that involve the direct use of an IP-based protocol and they use a common syntax for the server component.

server = [[userinfo "@"] hostport]

These URIs are mostly known from classic services such as the e-mail, sip and others. In these URIs the user information, if present, is followed by a commercial at-sign "@".

The URIs that do not make use of the slash "/" character for separating hierarchical components are considered opaque by the generic URI parser.

opaque_part = uric_no_slash *uric

uric_no_slash = unreserved | escaped | ";" | "?" | ":" | "@" | "&" | "=" | "+" | "$" | ","

Therefore, the URIs identify resources via the representation of their primary access mechanism (e.g., their network "location"). The schemes part within the URI provides the information about the access mechanism or protocols for accessing the resources. However, in certain cases there are alternative mechanisms

to access the resource, than the protocol indicated in the URI. Therefore, there are Gateways, proxies, caches, and name resolution services that are necessary to access some resources, independent of the protocol of their origin. The resolution of some URI may require the use of more than one protocol (e.g., both DNS and HTTP are typically used to access an "http" URI's resource).

The addressing in IP networks based on URI relies on a fixed infrastructure. DNS queries are made to obtain the destination IP address from the URI, either from the Fully Qualified Domain Name (FQDN) or from the telephone number (i.e. ENUM [14]). However, in order to provide seamless behavior over different access networks, the underlying layers have to provide an equivalent naming mechanism so that upper layers keep using URI addressing. This naming infrastructure widely used in fixed IP networks may not exist on new technologies such as Ad Hoc networks where there are no fixed infrastructure or DNS servers available.

2.1.2 Routing in IP networks

In IP networks the IP addresses or URIs identify the hosts or endpoints. The URI contains a host or domain part that identifies the destination and the corresponding IP address is obtained by querying DNS. The point of attachment of the host to the network may change and a single address may lead to different host locations depending on the new attachment. The sending host resolves addresses using the SRV, MX or CNAME records of DNS, described later in this section. The sending host may additionally use other protocols to determine the location, such as finger [16], rwhois [17], LDAP [18], multicast-based protocols or operating system dependent methods.

Thus, the first step is to map the destination URI into an IP address that is used as the *routing number*. Each host or endpoint can obtain its own IP address (es) statically or dynamically. The most common way is to obtain the IP address dynamically, using the Dynamic Host Configuration Protocol (DHCP) [19]. The host requests an IP address from the DHCP server that maintains a pool of addresses. The IPv6 protocol contains an inbuilt procedure for auto configuring

IPv6 addresses (neighbor discovery process). Ad Hoc nodes using IPv4 addresses also have to auto configure their addresses. They randomly select an IP address from a private address space, and using the ARP protocol the nodes test with their neighbors, whether the address is already taken or not.

Once the hosts have obtained their IP addresses, the routing protocols such as RIP [8], OSPF [9] and BGP [20] create the local routing table that is used for binding IP addresses with the communication interfaces. These protocols are part of the IP technology as they provide shortest routes among endpoints that implement the IP protocol.

6.1.2.1 Domain Name System

Domain Name System (DNS) [11] *is a hierarchical, distributed method of organizing the name space of the Internet.* It administratively groups hosts into a hierarchy of authority that allows addressing and other information to be widely distributed and maintained. Before DNS was taken into use, the mapping between host names (URI as described in the previous section) and IP addresses were centrally maintained in a file that had to be continuously updated on each host within the Internet.

The information is distributed among different servers that form the Domain Name System. The domain name space consists of the distributed database that contains the address information within the global Internet space. The domain name space is structured like a tree, where the root is denoted by '.' at the top and the depth is limited to 127 levels. Each node in the tree has a unique name (i.e. *domain name)* to identify its position in the database. Thus, the domain name describes the path from the node to the root of the tree.

The tree structure of the domain name space contains multiple sub-tree named domains that can be administered by different organizations. The names of those domains follow the same semantics as the domain name and they identify the position of the domain in the database. The Internet domain name space is mainly divided into seven top-level domains, named "com" (used for commercial

organizations), “org” (for non-commercial), “edu” (for education organizations), etc. Furthermore, each organization can break its domain into subdomains and delegate the responsibility of those subdomains to other organizations. The data associated with a domain name is contained in a Resource Record (RR) that is presented in a binary form and is stored in the servers that are involved in the DNS system.

The zone is a logical division defined in DNS specifications for implementing the distributed databases within each domain. A zone contains the domain names and their associated data (i.e. all the single node address information of a certain delegated area), but also the pointers to the delegated data in case the domain is broken into subdomains that are delegated elsewhere. The administrator of each zone uploads the information using a master file.

Figure 23 shows an example of the DNS hierarchy structure including domains, subdomains and zones. DNS provides enough flexibility for designing the architecture and the number of servers managing each domain and zone. Usually, the same server maintains each domain and the zones governed by that domain. Nevertheless, when the number of addresses in the zone is large, the zone has its own server for maintaining all the information. This facilitates the administration and limits the domain server to maintaining a smaller number of addresses and a link to the zone server. In this case, the domain server would only have a copy of the zone or it would cache the zone data but it does not attend the queries for addresses under the zone. In Figure 23, servers; “tele.netlab.hut.fi” and “keskus.netlab.hut.fi” maintain the information in the “Networking Lab”. The server “tele.netlab.hut.fi” maintains larger information than “keskus.netlab.hut.fi”, so it requires that a single server will attend the zone formed in the “netlab” domain and the addresses under “tele.netlab.hut.fi”. The “keskus.netlab.hut.fi” server maintains the “netlab” domain and a copy of the information in the zone administered by “tele.netlab.hut.fi”.

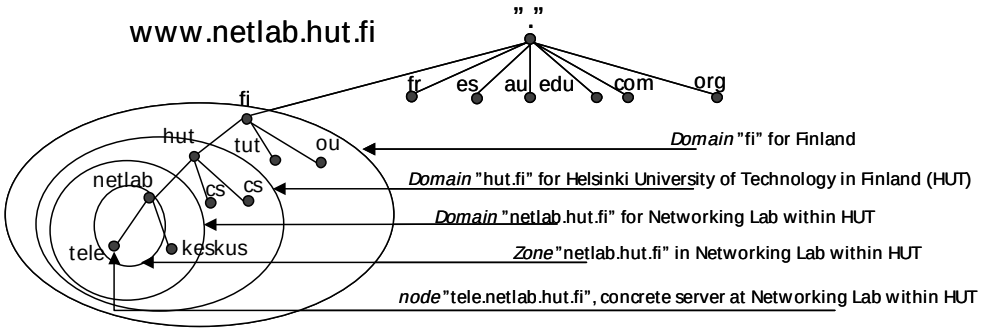

Figure 23. Domain and subdomain structures for "Networking Laboratory".

The addresses and all related information are stored in the zone file that is the database of the zone. The database file has different kinds of records that store all DNS related information. Table 3 lists the most common records stored in the zone file database.

Table 3. List of DNS record types.

Record Type	Description
SOA	Start Of Authority record indicates authority for this zone and provides the definition of the zone (e.g. Serial number, Refresh period, Retry period, TTL, Expire period).
NS	Lists a name server for this zone
A	Name-to-IP v4 address mapping
PTR	Address to name mapping
CNAME	Canonical name (for aliases)
MX	Enhanced mail routing Record
MF	Mail Forwarder address
MD	Mail destination address
X25	X.25 packet network address
ISDN	ISDN phone number address
RT	Route Through. It means that the traffic, which goes to the destination, should go through the address of the RT. E.g. through a proxy.
AAAA, A6	Name-to-IP v6 address mapping
NAPTR	Naming Authority Pointer, which contains the available ways of contacting the node identified by a specific name.

SRV	Record, which is used to map different services against an IP address and a port.
TXT	Textual representation to have a comment available
RP	Textual representation to present the responsible person of the server or service.

Some of these records as the one represented in Figure 24, are stored in the zone databases and they keep the addresses and other relevant information for the terminals within that zone.

	IN	NS	tele.netlab.hut.fi.
	IN	NS	keskus.netlab.hut.fi
Localhost	IN	A	127.0.0.1
tele	IN	A	192.189.249.2
Keskus	IN	A	192.189.249.3
Pc20			192.189.249.4
…….	……	…..	……..

Figure 24. Example of records stored in the zone database.

DNS works in the client-server mode where the *Name servers* constitute the server and resolvers constitute the client. The resolvers are normally system routines located in the terminals and receive the queries for mapping the URI into an addressable IP address. The resolver generates and sends a DNS query message to its default name server. The address of the default name server is configured in the resolver statically or is obtained using the DNS multicast discovery address. The resolver interprets the DNS responses and returns an answer to the program in the terminal that required the DNS service.

The Name server is the entity that responds to the query from the resolver using the information stored on its own database or requested from other distributed name servers. The Name server contains the authoritative information of one or more zones that it is responsible for but it also caches non-authoritative data of other

parts of the domain name space. Therefore, the information of any zone is available from several name servers. This mechanism guarantees the system resilience and reliability, since the information is available from multiple points. The name servers use specific synchronization mechanisms to keep information consistent and every DNS name server has a configuration file that defines all the zones that the server is authorized to reply to.

The DNS messages exchanged between the clients and the DNS servers are defined in RFC 1034 [10] and RFC 1035 [12]. The top-level format of the DNS message is divided into “Header”, “Question”, “Answer”, “Authority” and “Additional” sections. A query message only presents the “Header” and the “Question” sections. The “Header” section contains the control information (e.g. Message identifier, QR, OPCODE, RCODE, etc). The “QR” field is set for indicating that the message is a query. The “OPCODE” field specifies the type of the query. The “AA” field is included in the name server response for indicating that it is authoritative for the asked domain name. The “RD” and “RA” fields are used for recursion service. The “RCODE” field included in the response indicates the lookup status (e.g. no error, format error, server failure or name error, etc). The “Question” section contains a description of the type of query sent to the name server (e.g. domain name, type and class of query). The “Answer” section includes the records that contain the answer to the query. The “Authority” section contains records indicating other authoritative servers that may answer the query. The “Additional” section carries records that may be used for certain services. Figure 25 shows recursive DNS queries from the Name Server to resolve a name to an IP address requested by the resolver in the user device.

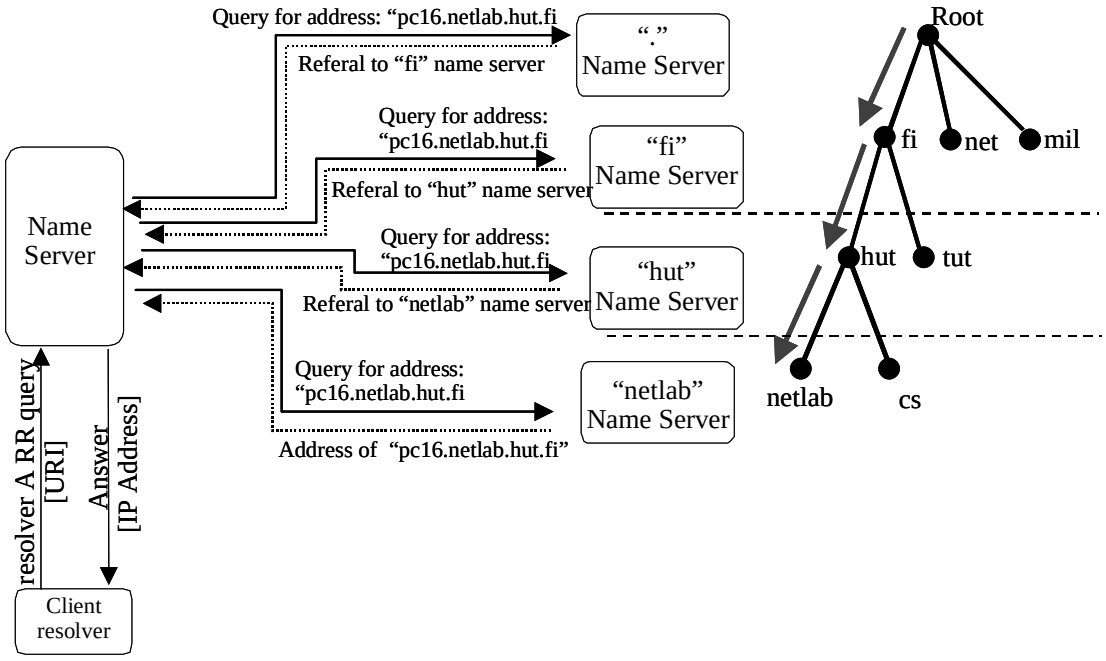

Figure 25. Example of DNS resolution name ("pc16.netlab.hut.fi") to IP address.

DNS services are the most versatile mechanism in wired and wireless IP networks to provide name (i.e. URI) to IP address translation as primary service but also IP addresses to name translation as secondary service [62],[63]. DNS also provides a service (i.e. ENUM [14]) to translate phone numbers to Fully Qualified Domain Names (FQDN) or URI and from there translate the name (i.e. FQDN) to IP address. Figure 26 shows the ENUM process for mapping a telephone number (i.e. E164 [86]) to a FQDN or URI. After obtaining the URI the user has to query again the DNS service to resolve the URI to an IP address. N: "i" in the number in figure.

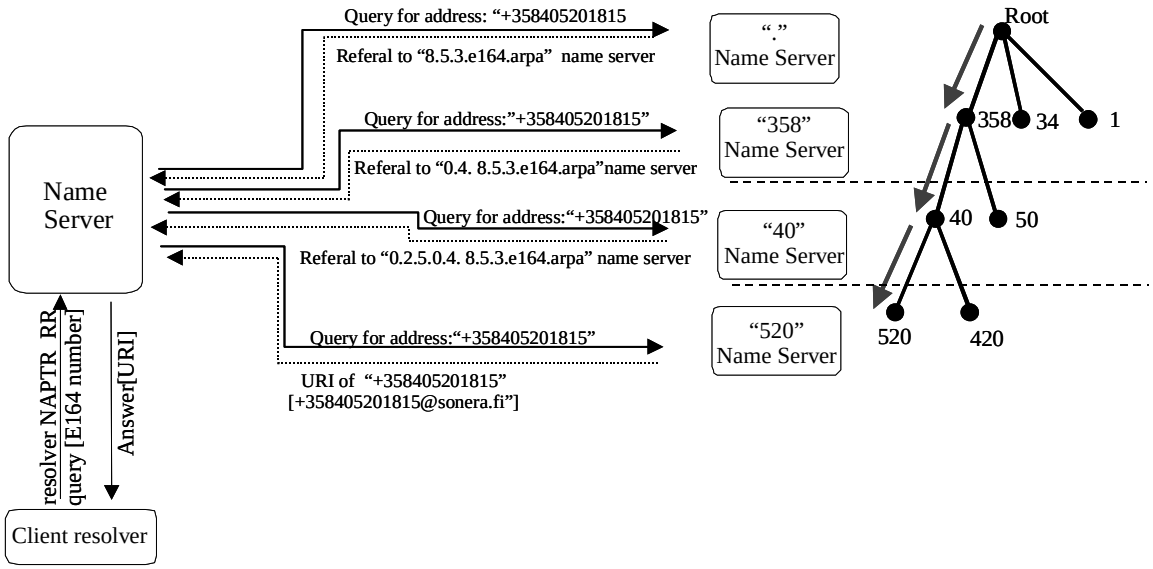

Figure 26. ENUM service to translate E.164 number to URI using DNS NAPTR service.

However, implementing DNS services in Ad Hoc networks is rather difficult since the nodes keep moving, they appear and disappear constantly and the DNS server should keep track of all the changes. The fact that Ad Hoc nodes often appear and disappear is an obstacle to reach them by their names. Therefore, DNS services would be used from Ad Hoc networks when accessing to hosts located in fixed IP networks. Thus, a service discovery mechanism to make the DNS servers located in the fixed IP networks visible from Ad Hoc networks would enable efficient addressing between Ad Hoc nodes and fixed IP nodes.

6.2 IP Telephony application

IP telephony is becoming the new leader in the telephony world. Day by day the number of companies interested in IP telephony is continuously increasing and new attractive services are in progress. Users demand new facilities and services that must be provided regardless the network technology underneath (SCN, fixed or wireless IP, Ad Hoc networks, etc). The Session Initiation Protocol (SIP) is the de facto signaling protocol to set up VoIP sessions.

"The Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying and terminating multimedia sessions or calls with one or more participants." [3]

In this chapter we focus on the study of the Session Initiation Protocol (SIP) [3] and give a general idea of the SIP protocol presenting its characteristics, the structure of the messages, and the behavior of the servers. Afterwards, the routing and addressing mechanism used by SIP are introduced. The aim of this chapter is to describe SIP and the additional location services and mobility tools utilized for the normal functioning and to discuss the possibility of adding new extensions.
SIP is the protocol that could provide mobility at session layer for Ad Hoc networks as depicted in Figure 27. In this example the Ad Hoc node including the SIP application moves from original point of attachment to a the new location where the connectivity is available via existing Ad Hoc network. The node finds the local SIP server and registers there. The registration is forwarded to the original network as described in 3G networks. This process allows that the node is reachable for other nodes at session layer that want to initiate a session with the node using SIP as described in message 3).

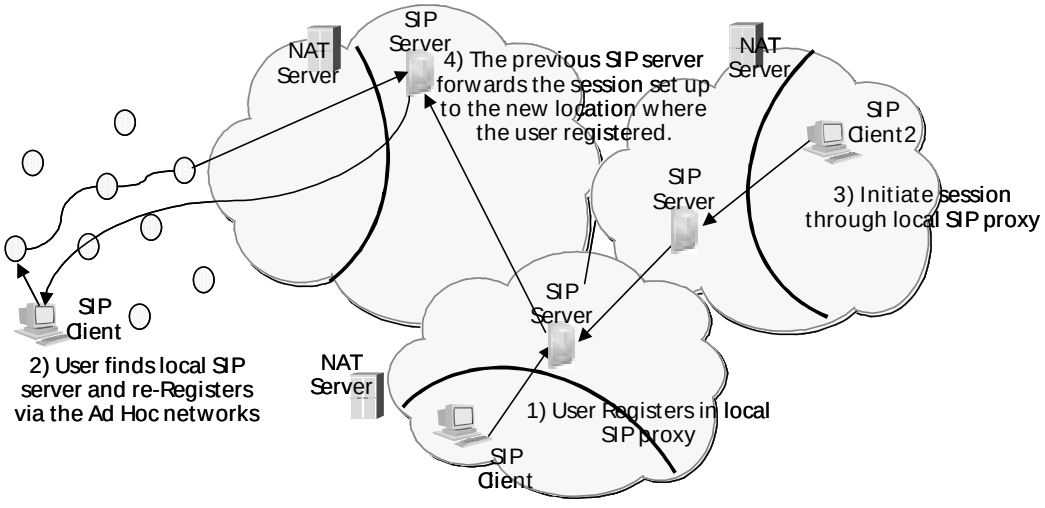

Figure 27 Connectivity at session layer with SIP.

Internet telephony requires a set of control protocols for connection establishment, capabilities exchange, and conference control. Initially, two standards emerged to meet this need. One is the ITU Recommendation H.323, and the other is the IETF Session Initiation Protocol (SIP) presented in this chapter. H.323 [31] is a rather complex protocol due to the hundreds of elements that it defines, and the use of several protocol components. The main drawback is no clean separation of these components. Thus, for deploying new services on top of H.323 usually requires interaction between several of those components to achieve a simple task. H.323 was the first signaling protocol for VoIP services in IP networks, thus most of the existing systems still use this protocol. On the other hand, SIP encodes its messages as text, similar to HTTP [37] and the Real Time Streaming Protocol [38]. This leads to simple parsing and generation, which allows easy extensibility. SIP is a rather new protocol adopted in wired and wireless systems [2], thus there are still few commercial systems deployed with this protocol. SIP is a good example of the Internet philosophy favoring a modular design where the result is the sum of multiple pieces that define a complete signaling protocol with the appropriate addressing, routing and mobility capabilities. These features, in addition to the flexibility that the protocol itself provides for adding new extensions and features needed for tailoring the protocol to different environments, define SIP as the right signaling protocol for IP telephony.

SIP is a quite modular protocol based on a set of headers each with a small number of values and parameters that contain more information. In addition to the message structure, SIP encompasses basic call signaling, user location, and registration. SIP itself provides an advanced signaling mechanism but it supports QoS, directory access, service discovery, session content description and conference control, which are orthogonal, and reside in separate protocols.

This set of advantages influenced the decision of adopting SIP as the signaling protocol when moving from 2nd generation (GSM [35]) into IP based networks in 3rd generation (UMTS [2]). H.323 continues in some products but SIP became the ‘de facto’ protocol for IP telephony in both fixed and wireless IP networks.

The next generation of wireless networks is aiming to bring advanced voice support into the data networking age, and the protocols must be targeted directly to the user services. The products should easily integrate into a real network with little or no modification to its underlying infrastructure (fixed IP, wireless IP, etc). Furthermore, the protocols should be easily extensible without breaking the existing implementations and these are the criteria that advocated SIP as the winner protocol in the voice and packet services creation.

6.1 SIP protocol

The Session Initiation Protocol (SIP) is a text-based protocol similarly to the Hyper Text Transport Protocol (HTTP). This allows easy implementation, easy debugging and makes SIP extensible and flexible. It is designed to be independent of the lower-layer transport protocol. This is because it has its own reliability system and can work with UDP as well as with TCP transport protocols.

SIP invites both persons and "robots" such as a media storage service. SIP can be used to initiate sessions as well as invite members to existing sessions. SIP invites parties to both unicast and multicast sessions. It can initiate multiparty calls using a Multipoint Control Unit (MCU) or fully meshed interconnection instead of multicast.

SIP does not offer conference control services and does not prescribe how a conference is to be managed, but it can be used to introduce conference control protocols. It can invite users to sessions with and without resource reservation.

SIP supports five facets of establishing and terminating multimedia communications:

- **User location**: determination of the end system to be used for communication.
- **User capabilities**: determination of the media and media parameters to be used.
- **User availability**: determination of the willingness of the called party to

engage in communications.

- **Call setup**: establishments of call parameters at both called and calling party.

- **Call handling**: including transfer and termination of calls.

SIP supports transparent name mapping and redirection services, allowing the implementation of ISDN and Intelligent Network telephone subscriber services that enable personal mobility. *Personal mobility* is defined as the ability of end users to originate and receive calls and access subscribed telecommunication services on any terminal in any location, and the ability of the network to identify end users as they move.

6.2.1 SIP components

There are two components in a SIP system: a User Agent (UA) and a network server. A UA is an end system that acts on behalf of a user. Usually it consists of two parts, a client (UAC) and a server (UAS), as the user probably is wishing to both be able to call and to be called. The UAC is used to initiate SIP requests while the UAS receives requests and returns responses on behalf of the user.

There are two kinds of network servers, namely, the proxy and the redirect servers.

A SIP proxy server forwards requests to the next server after deciding which one it should be. This next server could be any kind of SIP server and the proxy does not need to know the type of the next server. Before the request has reached the UAS it may have traversed several servers. Those will be traversed in reverse order by the response. A SIP proxy server can be stateful or stateless. Table 4 shows the differences between the two models. When stateful, a proxy remembers the incoming request, which generated outgoing requests. For that purpose the stateful server creates a new process to attend each new incoming request. Conversely, a stateless proxy sequentially processes each new request and forgets all the

information once an outgoing request is generated. It has the benefits of less processing and memory requirements in the server.

Table 4. Proxy server: stateful, stateless.

STATEFUL PROXY SERVER	STATELESS PROXY SERVER
Maintains call context	No call context
Replicates UAS/UAC to process requests/responses	Response is not based on UA replication
Call state and transaction state can be maintained	Provides client anonymity
Forking proxies require state	Restricted gateway access
TCP proxies must be stateful for reliability	High processing capacity
Enhanced services require state for execution	Allows for easier replication than the stateful model
Can populate billing information	Can have semi-stateful proxy for ultimate benefits

A redirect server does not forward requests to the next server. Instead of that, it sends a redirect response back to the client containing the address of the next server to contact with.

There is also a server that accepts REGISTER requests, which is called the registrar. A registrar is typically co-located with a proxy or redirect server and may also offer location services. The SIP Registrars can be located in Ad Hoc networks or in fixed networks. A SIP Registrars located in Ad Hoc networks can receive the REGISTER messages from SIP User Agents running in Ad Hoc nodes or REGISTER messages from SIP User Agents located in fixed networks. A SIP Registrars located in a fixed network can also receive REGISTER messages from nodes located in the Ad Hoc network or in the fixed infrastructure.

Therefore, the knowledge about available SIP Registrars is important in order to establish sessions between nodes irrespectively from where are they located. Information about SIP User Agents located in the fixed networks can be made

available to Ad Hoc nodes by discovering the SIP Registrar. In the same way, the SIP User Agents located in Ad Hoc nodes can be made available to the fixed infrastructure if they register in a SIP Registrar that is discoverable from nodes in fixed networks. The signalling messages will be forwarded by the Ad Hoc nodes as any other IP packets.

6.2.2 Basic protocol functionality and operation

To understand the basic operation of the SIP protocol, the following section presents some pictures showing the main transactions and how the different servers take part in the establishment of the desired communication between two UA.

6.2.2.1 SIP Invitation

A successful SIP invitation consists of two requests. The first one is always an *INVITE* and it is followed by an *ACK*. The *INVITE* request asks the callee to join a particular conference or to establish a two-party conversation. After the callee has agreed to participate in the call, the caller confirms that it has received that response by sending an *ACK* request. If the caller no longer wants to participate in the call, it sends a *BYE* request instead of an *ACK*. Figure 28 presents the basic transactions to set up a call.

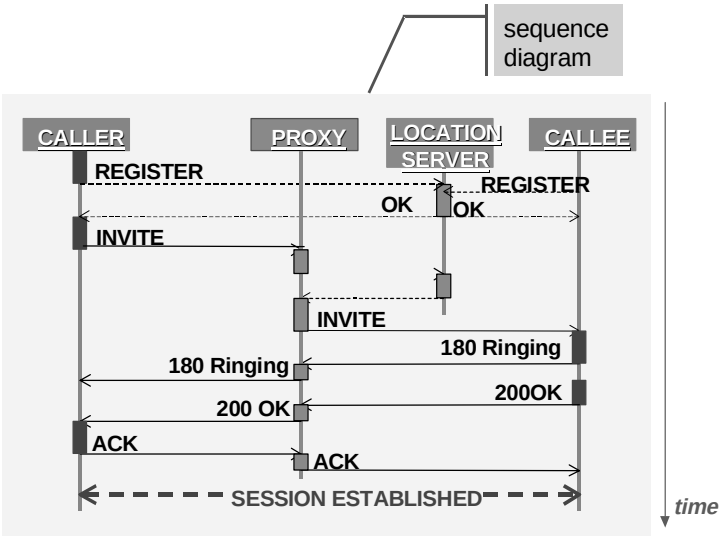

Figure 28. Call Setup (both endpoints registered, proxy routed call setup).

6.2.2.2 Locating a User

When a proxy or a redirect server contacts a location server, it can respond with a list of zero or more addresses where the user can be reached. This is because the user may be logged at more than one host or because the location server has inaccurate information. The action taken on receiving a list of locations varies with the type of the SIP server.

A redirect server will tell the caller all the addresses where the callee can be located and it is the caller, which decides what to do.

A proxy server will try the addresses given by the location server sequentially or in parallel until the call is successful or the callee has reclined the call.

To follow the way the message has taken, when a proxy server forwards a SIP request it adds a *VIA-header* to the message with its address. It is done so the response can follow the same way back. The order of the *VIA-headers* is important: a new *VIA-header* must be added to the end of the *VIA-headers* list of the message. This is also useful to prevent loops. A proxy server must not forward a request to a server that is already in the *VIA-header* list.

6.2.3 SIP Addresses

The *objects* addressed by SIP are users at hosts identified by a SIP URI, which has the form: "user@host" as described in previous chapters. The *user* part is a user name, a civil name or a telephone number. The *host* part is a domain name having e.g. a numeric network address.

A SIP address can designate an individual, the first available person from a group of individuals or a whole group. SIP URIs are used within SIP messages to indicate the originator, the current destination and the final recipient of a SIP request. Also they are used to specify redirection addresses, although this is not always true because some of the addresses mentioned above may be non-SIP URLs. SIP URLs can also be embedded in web pages or other hyperlinks indicating the use of the *INVITE* method.

6.2.4 SIP Services

As SIP is a text based Internet protocol, it is an open, distributed, and evolving entity. Everyone can contribute with new extensions that will be discussed in the IETF working group and after long discussion the proposed extensions may be approved and standardised. This feature allows deploying new services inbuilt on the protocol, using new extensions. The flexibility of SIP allows adding new headers and deploying new services such as presence. The presence service consists of obtaining the presence information of some users registered in the network. The implementation of this service requires new SIP methods (i.e. SUBSCRIBE and NOTIFY) and new headers (i.e. Event). Thus, the user can subscribe to the presence information of some friends and when the friends update the presence information (e.g. the friend registers in the network and changes the status from OFFLINE to ONLINE), the user will receive a notification indicating the change. SIP allows having these new services as an inbuilt feature by adding new methods and headers. In certain cases implementing these new services on mobile networks requires certain enhancements such as sending only the changes instead of all the presence information [41]. Figure 29 represents a flow diagram with basic messages for subscribing to the presence information of another user.

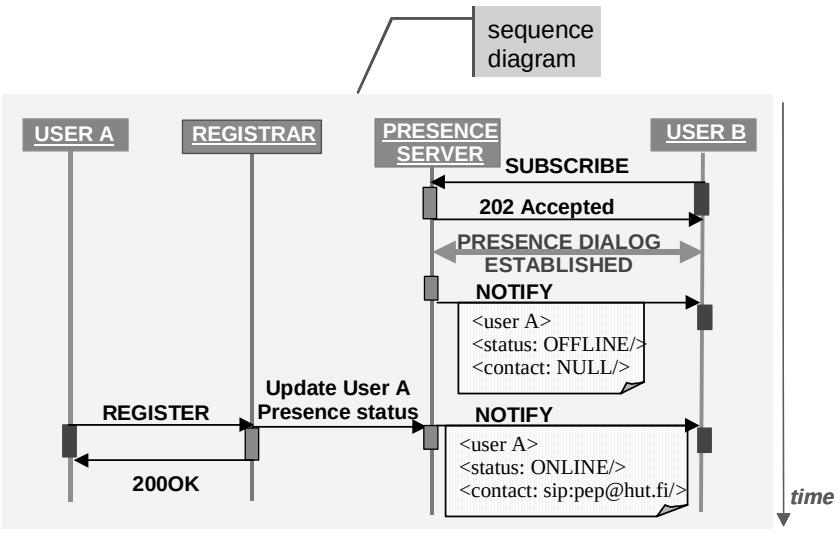

Figure 29. Presence flow.

In the area of personal mobility services, SIP provides rich support for this functionality as an inbuilt characteristic in the SIP User Agent and in the SIP proxies that may fork the request to multiple servers. This feature allows the deployment of Location based services (LBS) which are associated with the spatial or geographical locations of some entities (i.e. users or hosts) [42]. The key for achieving LBS consists of how a spatial location of an interested entity is collected, notified to the interested entities, and available upon request from service provisioning elements. SIP has the generic nature for managing media sessions as well as supporting network presence and messaging. SIP can also transport content data. It is therefore a protocol that can be used to carry spatial location information as the content data [43]. In addition, its infrastructure can be utilised for obtaining, storing, and facilitating that spatial location information to location-aware applications. The location based services obtain and register the location information associated with the user, which is required for enabling emergency calls [44] that are mandatory in telephony service. A flow example is depicted in Figure 30.

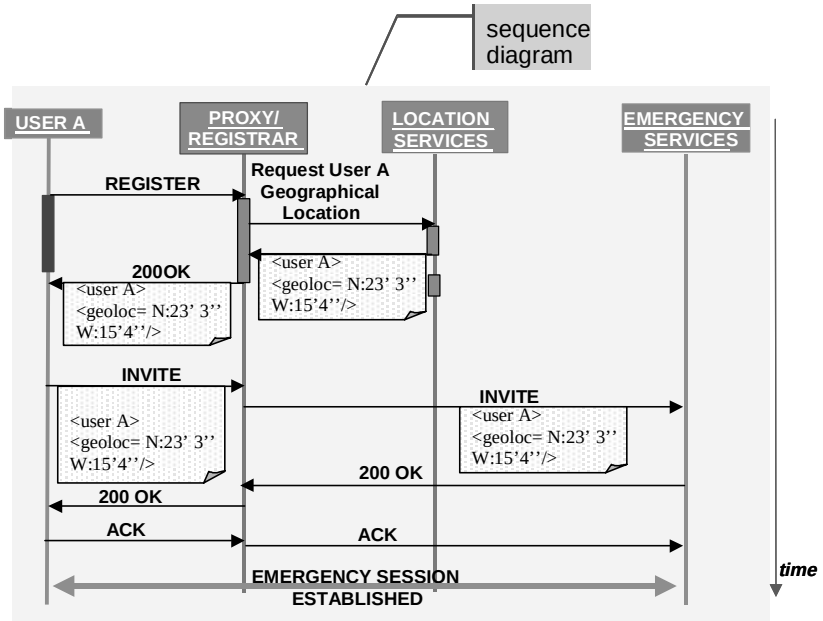

Figure 30. Location Services for SIP emergency sessions.

6.2 IP Telephony in Ad hoc framework test results

After implementing the Ad Hoc framework a validation testbed was deployed to demostrate the usage of Ad Hoc networks with real time applications such as VoIP using SIP protocol. The test cases were performed with iPAQs and laptops running the Ad Hoc framework [67]. The two main scenarios for the testbed where considering direct connection between nodes as shown in Figure 31 and a third node acting as sniffer to collect and analyse the exchanged data.

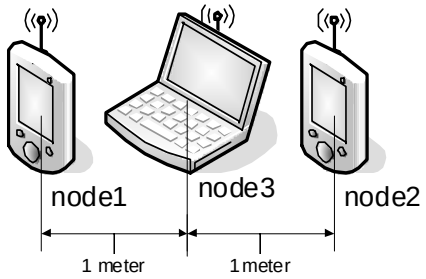

Figure 31. Test scenario 1 with direct connection between nodes.

The second scenario as shown in Figure 32 corresponds to the situation where there is a radio link between endpoints but there is also a fixed link between intermediate nodes. In this scenario the session is established between the two iPAQs acting as endpoint and the messages will traverse the fixed link between two intermediate nodes.

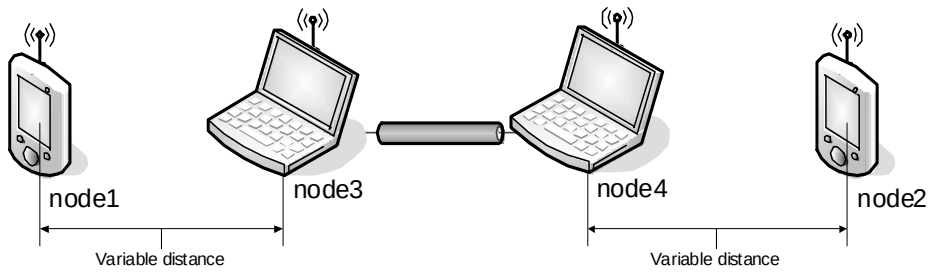

Figure 32. Test scenario 2 with wireless and wired links between nodes.

In all these scenarios the callee and caller devices contain SIP User Agents that contact the remote end using their IP address since at this point there are no DNS servers available. The intermediate nodes act merely as IP routers that forward the

SIP messages to the destination SIP User Agenet as they will forward any other IP message (e.g. HTTP request or response).

After implementing the proposed service discovery the nodes can find available SIP Registrar servers that store information about other registered SIP User Agents. Using the knowledge about SIP Registrars located in the fixed infrastructure but also Registrars located in the Ad Hoc netwok, the SIP User Agents can send the queries using SIP URI (e.g. “sip: jose@netlab.hut.fi”). If the Registrar is located in the fixed infrastructure and is discovered at the Ad Hoc network using the proposed service discovery, the User Agents located in the Ad Hoc networks can establish sessions with those User Agents registered in the Registrar located in the fixed networks. The results obtained from these scenarios demostrate that the Ad Hoc framework works properly in the different scenarios. The delay and packet loss increse in the second scenario but in overall the framework behaves as expected. Figure 33 shows the packet delay for the first scenario. The x-axis represents the same test with different different distances between the nodes (e.g. 1, 5 and 10m). The delay varies but it remains within 20ms range.

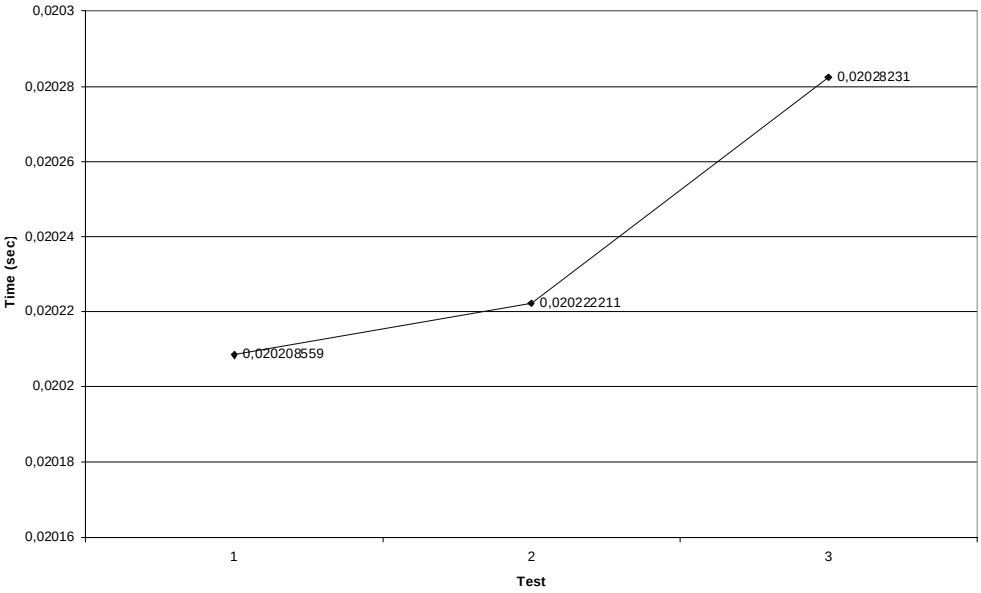

Figure 33. Mean delay for scenario 1.

Figure 34 shows similar results for the delay when considering scenario 2. X-axis represents the same test with different different distances between the nodes (e.g. 1, 5 and 10m). The delay varies but again it remains within 20ms range.

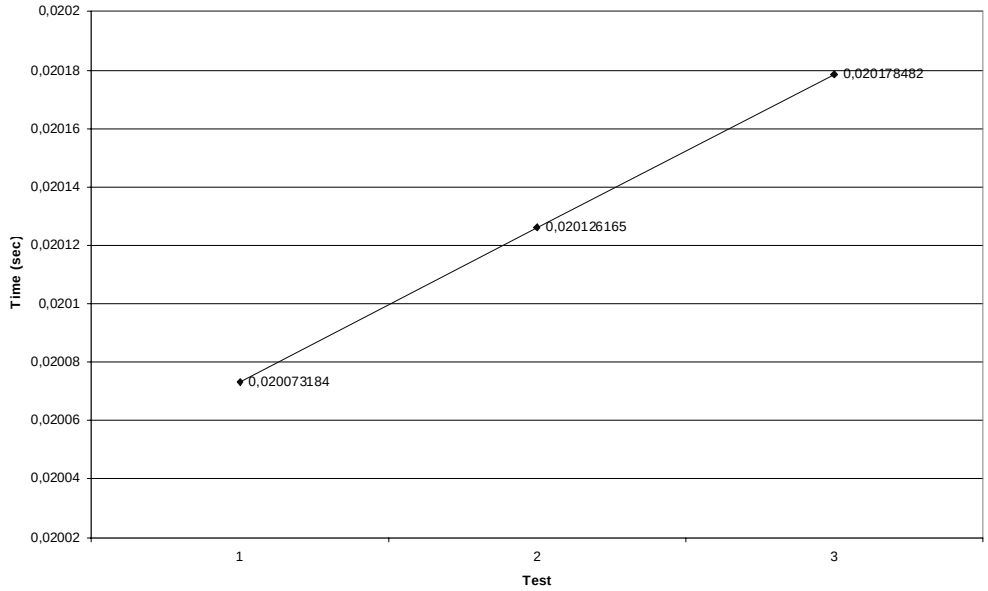

Figure 34. Mean Delay for scenario 2.

6.3 Conclusions

This chapter described the IP Telephony as the application used to validate the Ad Hoc framework implementation. The chapter includes a description of the IP Telephony service implemented using SIP as the signaling protocol for setting up multimedia sessions. The flexibility and extensibility of SIP were presented. SIP is becoming the preferred candidate for deploying telephony services over IP networks regardless of the technology (i.e. 3G, WLAN or Ad Hoc networks) and thus the suitable service to be deployed in Ad Hoc networks. The results of the testing demonstrate that the VoIP service can be implemented assuming a reasonable delay. This yields to the idea of having a set of Quality of service mechanism that will optimize the route discovery based on the quality requirements of the services. Thus, real time services such as VoIP require optimal routes with low delay in order to maintain the user experience from the fixed IP networks.

Chapter 7

Conclusions

This thesis presented the wireless IP networks and the routing and addressing mechanism in IP networks as basis to analyse the problems in routing and addressing in Ad Hoc networks.

Because of the extreme conditions typically of the Ad Hoc networks, it is envisioned that the specific conditions (i.e. network size, nodes mobility and density, etc) will determine the suitable routing protocol to be executed. Thus it is rather difficult to define an optimal protocol that suits for these continuously changing conditions. In small networks suits either reactive or proactive routing protocols while in medium to large scale networks and Hybrid protocol between proactive and reactive protocols should be used. To get acceptance the Ad Hoc networks should establish full connectivity with fixed IP networks. Therefore, a connectivity service should be provided and Ad Hoc networks should have a service discovery to achieve this connectivity. The service discovery procedure should be globally available within the Ad Hoc networks and should not impose any constraints to the nodes. The nodes with enough resources or willingness to contribute to the service distribution will provide the service discovery infrastructure to the other nodes with limited resources.

This thesis proposes a specific service layer structure to deploy a service infrastructure or “service backbone”. This service layer will enable using both proactive and reactive technologies in a seamless manner at lower layers. Two service backbone models were proposed (i.e. Hierarchical and Fully Distributed). Their advantages and drawbacks are presented in this thesis.

The hierarchical backbone defines a well-structured service layer that is independent from the underlying routing layer. This approach is based on the proposed on-demand algorithm (WCA) for the construction of the clusters. This constitutes a major advantage in comparison with other proactive clustering algorithms (such as LCA) which, introduce a significant overhead independently whether the nodes require any service or not.

Initial results obtained in simulations indicate that the use of an on-demand clustering algorithm for the construction of the backbone reduces considerably the overhead generated by service discovering. The time to discover a service also improves since the delay of service-control packets reduce as a consequence of less packet collisions and retransmissions. In scenarios with reduced mobility, once the backbone is defined, its maintenance is minimum, which allows an efficient distribution of services. Even in the case when mobility increases, the overhead to maintain the backbone is better in comparison to proactive clustering algorithms. Nevertheless, a possible drawback of this strategy is that a module for topology maintenance with WCA has to be defined in each node, instead of using the clustering information of certain routing protocols (such as GSR).

The distributed backbone is tightly related to the node taxonomy and needs the existence of "smart" nodes willing to provide the service layer, and nodes that require service provisioning. Thus, the "smart" nodes should have certain motivation for participating in the creation of the "service backbone" and thus contribute to the Ad Hoc network (e.g. service provisioning with charging requirements, operator support for accessing the core network services (UMTS), the distributed backbone implicitly implements the basis to provide full connectivity with fixed IP network, etc.). This method provides a non-reliable "service backbone" based on best-effort approach depending on the contribution from some "smart" nodes. Thus, the service provisioning is moved to a different layer (i.e. application layer or middleware provisioning) where depending on the interest from certain nodes the "service infrastructure" will be automatically built or not. This approach requires a cross layer architecture where the service discovery is linked with the routing layer. Thus, all nodes would have a service

layer where the service discovery will trigger the specific routing functionality built for this purpose (i.e. AODV service extension or service information in link state protocols). The main point of this proposal is that nodes with low resources (i.e. dummy) do benefit from the service layer without additional constraints to their normal behaviour.

This thesis proposes the service layer and the interaction with the routing protocols as mechanisms to access services in fixed networks. This implicitly enables full connectivity from Ad Hoc networks to fixed infrastructure. This approach uses reactive mechanisms to find a service (client sends requests for services and try to get replies from servers), instead of proactive mechanisms (servers announce their services to all the network). The proactive service discovering has the advantage of rapid availability of service information, but it does not scale well beyond local domains. In contrast, reactivity in service discovering consumes little traffic. Thus, proactive mechanism is used for sharing the service information among the "smart" nodes so the knowledge about available services in the networks is fully distributed.

This proposal is implemented in a test bed (i.e. Ad Hoc framework) to verify the design. IP telephony service is also implemented in the test bed to validate the proposal. The results demonstrate the feasibility of having demanding services such as telephony, regardless the network technology underneath (SCN, fixed or wireless IP, Ad Hoc networks, etc).

References

[1] The official IETF working group Manet webpage, <URL: http://www.ietf.org/html.charters/manet-charter.html>.

[2] The UMTS Forum , <URL: http://www.umts-forum.org/>

[3] M. Handley, H. Schulzrinne, E. Schooler and J. Rosenberg, "SIP: Session Initiation Protocol," *Request for Comments 3261*, *Internet Engineering Task Force*.

[4] Multiparty Multimedia Session Control (MMUSIC), online <URL: http://www.ietf.org/html.charters/mmusic-charter.html>.

[5] Institute of Electrical and Electronics Engineers, Inc., Short Description of the Standard, 1999. <URL: http://grouper.ieee.org/groups/802/11/main.html>

[6] S. Floyd, "General Architectural and Policy Considerations", RFC 3426, IAB, November 2002. <URL: ftp://ftp.rfc-editor.org/in-notes/rfc3426.txt>

[7] James Kempf, Rob Austein, "The Rise of the Middle and the Future of End to End: Reflections on the Evolution of the Internet Architecture", IAB, January 2003, draft-iab-e2e-futures-00.txt.

[8] C. Hedrick, "Routing Information Protocol", RFC 1058, June 1998. <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1058.txt>.

[9] J. Moy, "OSPF Version 2", RFC 1583, Obsoletes: 1247 March 1994, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1583.txt>.

[10] Y. Rekhter, T.J. Watson Research Center and T. Li, "A Border Gateway Protocol 4 (BGP-4)",RFC:1771, Obsoletes: 1654, March 1995. <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1771.txt>.

[11] P. Mockapetris, "Domain Names – Concepts and Facilities", RFC 1034, Obsoletes: RFCs 882, 883, 973, November 1987, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1034.txt>.

[12] P. Mockapetris, "Domain Names – Implementation and Specification", RFC 1035, Obsoletes: RFCs 882, 883, 973, November 1987, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1035.txt>.

[13] T. Berners-Lee, R. Fielding, U.C. Irvine and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax",RFC: 2396, Updates: 1808, 1738, August 1998, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc2396.txt>.

[14] Telephone Number Mapping (enum), IETF enum workgroup, Feb 2000 <URL: http://www.ietf.org/html.charters/enum-charter.html>.

[15] T. Berners-Lee, “Universal Resource Identifiers in WWW, A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web”, RFC 1630, June 1994, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1630.txt>.

[16] D. Zimmerman, “The Finger User Information Protocol”, RFC 1288, Obsoletes: RFCs 1196, 1194, 742, December 1991, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1288.txt>

[17] S. Williamson, M. Kosters, D. Blacka, J. Singh and K. Zeilstra, “Referral Whois (RWhois) Protocol V1.5“,RFC: 2167, Obsoletes: RFC 1714, June 1997, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc2167.txt>.

[18] W. Yeong, T. Howes and S. Kille “Lightweight Directory Access Protocol”, RFC 1777, Obsoletes: RFC 1487, March 1995, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1777.txt>.

[19] R. Droms, "DHCP: Dynamic Host Configuration Protocol", RFC 2131, Obsoletes: 1541, March 1997, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc2131.txt>.

[20] Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4)”, RFC 1771, Obsoletes: 1654, March 1995, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1771.txt>.

[21] Plummer, D. "An Ethernet Address Resolution Protocol", RFC-826, November 1982, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc826.txt>.

[22] J. Postel, “Internet Protocol Darpa Internet Program Protocol Specification”, STD 5, September 1981, <URL: ftp://ftp.rfc-editor.org/in-notes/std/std5.txt>.

[23] The Internet Engineering Task Force, online <URL: http://www.ietf.org/>.

[24] G. Malkin, “RIP Version 2 Carrying Additional Information”, RFC 1388, Updates: RFC 1058, January 1993, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1388.txt>.

[25] G. Malkin, “RIP Version 2 Carrying Additional Information”, RFC 1723, Obsoletes: 1388, Updates: 1058, November 1994, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc1723.txt>.

[26] E. Rosen, “Exterior Gateway Protocol (EGP)”, RFC 827, O October 1982, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc827.txt>.

[27] R. Kantola, J. Costa-Requena, Nicklas Beijar, “SIP 2000 beyond H.323”, Paris, September 15-17, 1999, <URL: http://www.upperside.fr/bamgc.htm>

[28] R. Kantola and J M Costa-Requena: "The Server Cache Synchronization Protocol - a component for Directory Enabled Networks", SPIE VVD'99.

[29] J. Costa-Requena, Ignacio Gonzalez Olías, Raimo Kantola, Nicklas Beijar: "Autoconfiguration mechanism for IP Telephony Location Servers". IFIP, TC6, WG6.2, Proceedings of the 6 International Symposium on Communication Networking, 2002, Perth, Western Australia. <URL:http://www.ee.uwa.edu.au/~ifip2002/>

[30] R. Kantola, Jose Costa-Requena, Nicklas Beijar, "An Architecture for an SCN/IP Telephony Routing Testbed", Proceedings of The first IP Telephony Workshop, Berlin, Germany, April 2000. <URL: http://www.fokus.gmd.de/research/cc/glone/projects/iptel2000/ pg.php3>.

[31] International Telecommunication Union, *H.323 recommendation*, online <URL: http://www.itu.int/itudoc/itu-t/rec/h/h323.html>

[32] Internet Assigned Numbers Authority, <URL: http://www.iana.org/>.

[33] R. Kantola, Jose Costa-Requena, Nicklas Beijar, "A Common Numbering Infrastructure for IN and IP Telephony", IN2000, Cape Town, South Africa, May 2000. <URL: http://www.comsoc.org/IN2000/>

[34] R. Kantola, Jose Costa-Requena, Nicklas Beijar, "Interoperable routing for IN and IP Telephony", Computer Networks, Volume 35, Issue 5, pp. 597-609, April 2001. <URL:http://www.elsevier.com>.

[35] 3GPP Specifications Home Page (GSM (including GPRS and EDGE) and W-CDMA specifications, 3G specifications: UTRAN, UMTS (in Europe) and FOMA (in Japan). <URL: http://www.3gpp.org/specs/specs.htm>.

[36] J. Costa-Requena and I Espigares: "Security concerns in 3G networks", The Internet and Multimedia Systems and Applications (IMSA2001). Proceedings of the IASTED International Conference, pp 91-95, August 2001.

[37] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, Obsoletes RFC2068, June 1999, <URL: ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt>.

[38] H. Schulzrinne, R. Lanphier and A. Rao, "RTSP: Real Time Streaming Protocol," RFC 2326, April. 1998,<URL: ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt>.

[39] M. Handley and V. Jacobson, "SDP: Session Description Protocol," *Request for Comments 2327, Internet Engineering Task Force*, Apr. 1998.

[40] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *Request for Comments 1889, Internet Engineering Task Force*, Jan. 1996.

[41] M. Lonnfors, J. Costa-Requena, E. Leppanen and H. Khartabil, "Session Initiation Protocol (SIP) extension for Partial Notification of Presence

Information", SIMPLE WG, Internet-Draft last Call, April 2004, <URL: http://www.ietf.org/internet-drafts/draft-ietf-simple-partial-notify-02.txt>.

[42] H Tang and J Costa-Requena: "Serving Spatial Location over Internet". Proceedings of the ACM 2nd International Conference on Mobile Data Management, Springer, pp.246-251, Jan. 2001.

[43] J. Costa-Requena and H Tang:"Enable SIP with Spatial Location for Emergency Call Services",IEEE ICCCN2001, October 2001.

[44] J. Costa-Requena and Haitao Tang, "Application of Spatial Location Information to SIP", Cluster Computing Journal. <URL: http://www.baltzer.nl/cluster/cluster.asp>.

[45] J. Costa Requena, Raimo Kantola, Nicklas Beijar: Mobility and Network Management in Ad Hoc Networks, IASTED International Conference, Communication Systems and Networks (CSN 2002), September 9-12, 2002, Málaga, Spain. <URL: http://www.iasted.org/conferences/2002/spain/csn.htm>.

[46] Sun Microsystems, Inc, "JiniTM Architecture Specification", Version 1.2, <URL: http://www.sun.com/jini/. December 2001>.

[47] Universal Plug and Play Forum, "Universal Plug and Play Technology UPnP", <URL: http://www.upnp.org/>.

[48] Salutation Consortium, "The Application Programmer's Interface of the Salutation Architecture ", <URL: http://www.salutation.org/>.

[49] Open Services Gateway Initiative (OSGi), "OSGi - The Managed Services Specification", <URL: http://www.osgi.org/>.

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

[51] J. Costa Requena, Nicklas Beijar, Raimo Kantola, "Replication of Routing Tables for Mobility Management in Ad Hoc Networks", Med-hoc-net 2002, September 4-6, 2002, Chia, Italy. <URL: http://www-rp.lip6.fr/medhocnet/>

[52] J. Costa Requena, Nicklas Beijar, Raimo Kantola,"Replication of Routing Tables for Mobility Management in Ad Hoc Networks", ACM Wireless Networks (WINET) Journal, 2003.

[53] O. Arpacioglu, T. Small and Z.J. Haas, "Notes on scalability of Wireless Ad Hoc networks", IETF Internet Draft "draft-irtf-and-scalability-notes-00.txt", August 2003.

[54] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Lanouiti, L. Viennot and T. Clausen, IETF MANET Internet Draft "draft-ietf-MANET-olsr-02.txt", July 2000

[55] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol", IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks, vol 17, No 8, pp. 1395-1414, August 1999.

[56] C. E. Perkins and E.M. Royer, "Ad-hoc On Demand Distance Vector Routing", Second IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, February 1999.

[57] IEEE 802.11a, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications <URL: http://standards.ieee.org/reading/ieee/std/lanman/>.

[58] AODV-UU, Ad Hoc On Demand Distance Vector implementation created at Uppsala Universtity. <URL: http://user.it.uu.se/~henrikl/aodv/>

[59] BOOTP protocol <URL: http://www.ietf.org/rfc/rfc0951.txt?number=951>

[60] I. Espigares, Master Thesis: Implementation of the Internet Call Waiting Service using SIP, Helsinki University of Technology, Networking Laboratory, 1999.

[61] J. Yebenes, Master Thesis: A Scalability Analysis of the Server Cache Synchronization Protocol (SCSP), Helsinki University of Technology, Networking Laboratory, 2000.

[62] X. Zhen, Master Thesis: Scalability Analysis of ENUM for IP Telephony Routing, Helsinki University of Technology, Networking Laboratory, 2001.

[63] A. Heino, Master Thesis: ENUM Service in the 3G Networks, Helsinki University of Technology, Networking Laboratory, 2002.

[64] I. Gonzalez Olias, Master Thesis: Security and auto-configuration of Location Servers for IP Telephony, Helsinki University of Technology, Networking Laboratory, 2002.

[65] L. Xiao, Master Thesis: Ad Hoc Routing Framework design and implementation, Helsinki University of Technology, Networking Laboratory, 2003.

[66] J. Gutierrez Plaza, Master Thesis: Design and implementation of OLSR protocol in an Ad Hoc Routing Framework, Helsinki University of Technology, Networking Laboratory, 2003.

[67] J. Garcia Sanchez, Laboratory Report: A Study of VoIP over Wireless Local Area Network, Helsinki University of Technology, Networking Laboratory, 2003.

[68] Bluetooth: <URL: http://www.bluetooth.org>

[69] Mingliang Jiang, Jinyang Li and Y.C. Tay," Cluster Based Routing Protocol", IETF Internet Draft (work in progress), 1999, <URL: http://community.roxen.com/developers/idocs/drafts/draft-ietf-manet-cbrp-spec-01.txt>.

[70] T.W. Chen and M. Gerla, Global State Routing: A new routing scheme for Ad Hoc Wireless Networks, Proceedings of IEEE ICC, Atlanta GA, pages 171-175,1998.

[71] M. Gerla and J. Tsai," Multicluster, Mobile, Multimedia Radio Networks", Journal Wireless Networks, vol 1-3, pages 255-265,1995.

[72] Boris Mitelman and Arkady Zaslavsky, "Link State Routing Protocol with Cluster Based Flooding for Mobile Ad-hoc Computer Networks", Proceedings of the Workshop on Computer Science and Information Technologies (CSIT), 1999, Moscow Russia.

[73] Mainak Chatterjee and Sajal K. Das and Damla Turgut," WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks", <URL: citeseer.nj.nec.com/chatterjee01wca.html>.

[74] GloMoSim: Global Mobile Information Systems Simulation Library, <URL: http://pcl.cs.ucla.edu/ projects/glomosim/>.

[75] E.M. Belding-Royer, C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", IEEE Personal Communications Magazine, April 1999, pp. 46-55.

[76] C.E. Perkins, "Ad Hoc Networking", Addison-Wesley, Reading, MA, 2000] and [ER03: Elizabeth Belding-Royer, "Routing approaches in Mobile Ad Hoc Networks", in Mobile Ad Hoc Networking,

[77] Elizabeth Belding-Royer, "Routing approaches in Mobile Ad Hoc Networks", in Mobile Ad Hoc Networking, S. Basagni, M. Conti, S. Giordano, I. Stojmenovic (Editors), IEEE Press and John Wiley and Sons, Inc., New York, 2003.

[78] P. Jacquet, P. Muhlethaler, A. Qayyum, "Optimized Link State Routing Protocol", Internet Draft, draft-ietfmanet-olsr-00.txt, November 1998.

[79] B. Bellur, R. G. Ogier, F. L. Templin, "Topology Broadcast Based on Reverse-Path Forwarding (TBRPF)", IETF Internet Draft, draft-ietf-manet-tbrpf-01.txt, March 2001.

[80] D. B. Johnson, D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks", Mobile Computing, T. Imielinski and H. Korth (eds.), Kluwer Academic Publishers, pp. 153--181,1996.

[81] APE: Ad hoc Protocol Evaluation testbed. Department of Computer Systems at Uppsala, Sweden. <URL: http://apetestbed.sourceforge.net/>

[82] H. Lundgren, E. Nordstron, C. Tschudin, “Coping with Communication Gray Zones in IEEE 802.11 based Ad Hoc Networks”, Proceedings of the ACM Workshop on Mobile Multimedia (WoWMoM 2002), Atlanta (GA), September 28, 2002, pp. 49-55.

[83] MANET Meeting Report at 55th IETF Meeting in Altanta, Georgia USA <URL: http://www.ietf.org/proceedings/02nov/177.htm>

[84] Josh Broch, David A. Maltz, David B. Johnson, “Quantitative Lessons From a Full-Scale Multi-Hop Wireless Ad Hoc Network Testbed”, Proceedings of the IEEE Wireless Communications and Network Conference 2000 (WCNC 2000).

[85] S. Thomson and T. Narten,”IPv6 Stateless Address Autoconfiguration”. RFC 2462, IETF, Jun 1999.http://www.ietf.org/rfc/rfc2462.txt.

[86] Numbering plans guild E.164: The international public telecommunication numbering plan], International numbering plans, May 2000,<URL: http://www.numberingplans.com/index.php3>

[87] D. De Col, “Routing protocols for wireless ad hoc networks: performance evaluation of AODV and DSR”, Computer Science Laura Thesis, University of Pisa, October 2002 (in Italian).

[88] L. Kleinrock, K. Stevens, “Fisheye: A Lenslike Computer Display Transformation”, Technical Report, UCLA, Computer Science Department, 1971.

[89] Jaehoon Paul Jeong, Jungsoo Park, Hyoungjun Kim and Dongkyun Kim “Ad Hoc IP Address Autoconfiguration”. IETF draft. <URL: http://www.ietf.org/internet-drafts/draft-jeong-adhoc-ip-addr-autoconf-02.txt>. February 04.

[90] C. Perkins, “IP Mobility Support for IPv4”, RFC3344 IETF Proposed Standard , <URL:ftp://ftp.rfc-editor.org/in-notes/rfc3344.txt>,August 2002.

[91] P. Srisuresh, K. Egevang, “Traditional IP Network Address Translator (Traditional NAT)”, <URL:ftp://ftp.rfc-editor.org/in-notes/rfc3022.txt>, RFC3022 IETF Information, January 2001.