Architecture de communication mobile avec qualité de service

Jose Antonio Garcia Macias

Institut National Polytechnique de Grenoble

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

***Informatique :* Systèmes et Communications**

préparée au Laboratoire Logiciels, Systèmes, Réseaux (LSR)

présentée et soutenue publiquement

par

José Antonio García Macías

le 8 janvier 2002

Architecture de Communication Mobile avec Qualité de Service

(Mobile Communication Architecture with Quality of Service)

Directeur de thèse :

M. Andrzej Duda (LSR)

JURY :

M. Paul Jacquet	Président
M. Eric Horlait	Rapporteur
M. Bernard Tourancheau	Rapporteur
M. Claude Castelluccia	Examinateur
M. Andrzej Duda	Directeur de thèse

Vitae, non scholae discimus

Acknowledgements

Il n'y a guère au monde un plus bel excès que celui de la reconnaissance.

– J. de la Bruyère.

I am very grateful to all the people who, directly or indirectly, helped me during a long and bumpy road.

I'm indebted to Andrzej Duda, my thesis director, an intelligent person from whom a lot of things can be learned. I would like to thank the members of the jury: Paul Jacquet, Claude Castellucia, Eric Horlait and Bernard Tourancheau, for accepting to examine my work. Special thanks go to Mr. Tourancheau and Horlait for taking time out of their busy agendas to read my manuscript and give me useful feedback.

Without the invaluable help of Franck Rousseau and Gilles Berger-Sabbatel, my thesis work would have taken a lot longer; they coded most of the systems that were used for the implementation and testing phases related to this work, and also assisted me with the numerous technical problems that I had. Dominique (Pancho) Decouchant has been always ready to help me, not only at the academic but also at the personal level. Leyla Toumi has shared with me the ups and downs that come with the preparation of a Ph.D.; she has been a real good friend along the way. *Chokran!* My deepest thanks go also to Luciano Garcia, with whom I have shared not only good and bad times but also the apartment we used to rent together *¡Muchas gracias!*

I would also like to mention here my family, particularly my mother and my brothers, who have always loved me and believed in me. The staff at Praxis Telecom and the Teleddes Foundation have always showed me great support, specially Dr. Arturo Serrano, truly one of my role models, and my dear *big sister* Patricia Ramonetti.

Without the financial support of the Mexican Consejo Nacional de Ciencia y Tecnología (CONACYT) I would have never had the opportunity of living such memorable experiences as I have since my arrival in France, and I am not only talking from an academic point of view.

While at the LSR laboratory, I had the privilege of interacting with great people that have made my Ph.D. experience more educational, more rich, and more fun: Patricia Serrano, José Luis Zecchinelli, Genoveva Vargas, Edgard Benitez, Stéphane Drapeau, Elizabeth Perez, Claudia Roncancio, Khalid Belhajjame, Rafael Lozano, Marlon Dumas,

Olivier Lobry, Stéphane Lo-Presti, Christiane Plumere, Patrice Uvietta, Beatriz González, Raul García, François Challier, Sonia Nogueira, Liliane Di-Giacomo, Martine Pernice, Solange Roche, ... I have a really bad memory, so I'm surely not mentioning by name all the people I should. My head may forget, but my heart will not.

Table of Contents

List of Tables

List of Figures

Introduction

Chapter 1

Introduction

If we knew what it was we were doing, it would not be called research, would it?

— Albert Einstein

1.1 Motivation

The sudden and unexpected rise of the Internet into the commercial world in the mid-1990s catapulted the resurgence of networks using the Internet Protocol (IP). Nowadays, it is undeniable that IP networks are here to stay and that will continue to be relevant for years to come. One feature of IP networks that has undoubtedly played an important role in their longevity and relevance is their adaptability to unpredictable change. Indeed, end-to-end arguments [129] that move intelligence and new features away from the network's core and out to the edges, have constituted the main design principles and the philosophical guidelines for the evolution of IP and related protocols[1]. Contrast this approach of "stupid networks" [68] to that of "intelligent networks", such as the Public Switched Telephone Network (PSTN), which are inflexible and incapable to accommodate rapid change and incorporate new technologies [140].

Among the myriad of new technologies that have been incorporated in IP networks are wireless and mobile technologies. It is now commonplace, for instance, to find portable computers connected to the Internet by means of a wireless connection, in most cases provided by a Wireless Local Area Network (WLAN). These wireless links are not without problems, though, as they are very sensitive to interferences, signal quality decreases as a mobile moves away from its access point, and transmission speeds are not always as high as those offered by wireline links.

Coupled with the use of new technologies such as wireless networks is the emergence of new bandwidth-consuming, delay-sensitive applications: real-time multimedia communi-

[1] As commercial interests take over the Internet [64], many fear that end-to-end arguments will soon be no longer appropriate and have to be revised [20].

cations, networked games, distributed collaborative applications, peer-to-peer multimedia tools, immersive worlds, and others yet to be invented. All this means new problems, as such applications demand a quality of service (QoS) that goes beyond the best-effort service typically offered by IP networks.

1.2 Problem Description

It is then clear that the combination of local mobility in wireless IP networks, where QoS support needs to be in place due to ever more demanding applications, rises several challenges.

Do current wireless local area networks provide acceptable performances to guarantee QoS? Are IP mobility proposals such as Mobile IP good enough for handling local mobility? If not, what are the requirements for efficiently handling mobility in local environments? What is the relation (if any) between local mobility and QoS provision? Do current local mobility proposals take into account QoS support? What would a good model for supporting QoS be in wireless IP access networks? This and many others are questions we tried to answer.

We tried to find answers to these questions, always aiming for simplicity and having in mind the philosophy of the Internet of pushing new functionalities to the edges, making them transparent to the core. We proposed solutions that address the local mobility and QoS support aspects, aiming at a smooth integration of both. These solutions were prototyped and tested in order to verify their feasibility and performance; the implementation of our prototype is the result of an intense group effort within the Drakkar research group. All this work was carried out at the ENSIMAG, in LSR laboratory's facilities; development was made in the framework of the @IRS Project [1], a French national project for next-generation networks.

1.3 Document Organization

Globally, the organization of this document is as follows: in this introduction we have given an overview of the problems we are trying to solve; the presentation of the background and state-of-the-art is made in Chapters 2, 3 and 4; our proposed solutions are discussed in Chapter 5, and Chapter 6 presents prototypes and tests to validate some of these propositions. We finish by giving some conclusions and outlining perspectives for future work. Figure 1.1 gives a visual representation of how this document is organized. We will now proceed to describe with more detail what each chapter discusses.

Since our work deals with mobility and QoS management in wireless local area networks (WLANs), we introduce these types of networks making a particular emphasis on the IEEE 802.11 standard, since it is the type of network we used for our prototypes and tests. Nonetheless, we delve into enough detail during the discussion of other similar (and sometimes competing) WLAN technologies. A brief overview is given to discuss wireless technologies that may complement these WLANs in the wide area.

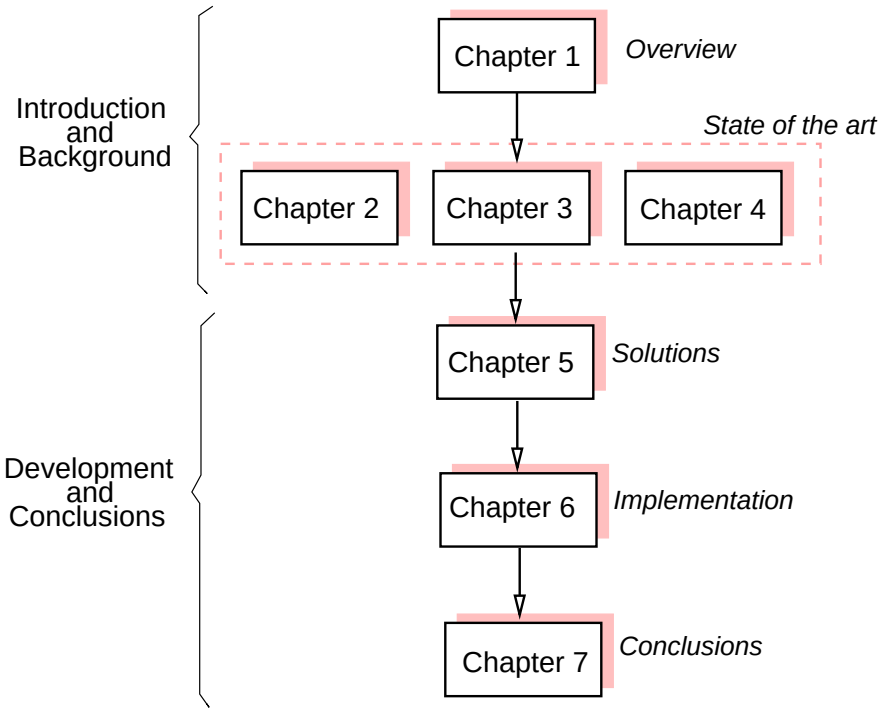

Figure 1.1: Organization of this thesis document.

Chapter 3 starts giving an overview of the concept of Quality of Service (QoS) in computer networks; then, the discussion is focused on QoS in the context of IP-based networks since that is the domain where our work is situated. We discuss the most prominent approaches for QoS in the IP world: integrated services (Intserv) and differentiated services (Diffserv).

The concept of mobility in IP networks is discussed in Chapter 4. We focus on the case of micro-mobility, discussing such approaches as Hawaii and Cellular IP. The idea of using the concept of in-band signaling for optimizing mobility management is then introduced. The chapter ends by presenting current on-going work —notably within the Internet Engineering Task Force (IETF)— in the field of seamless mobility.

In Chapter 5 we present our solutions for integrating mobility and QoS management. We start by separately presenting our mobility management solutions, then our QoS management solutions (based on the adoption of the Diffserv approach for providing QoS within the realm of wireless local networks), and then elaborate on the integration of both solutions by integrating out-of-band and in-band protocols.

Our propositions are validated by the implementation of prototypes and experiments carried out with them, as explained in Chapter 6. We start by introducing our experimental platform and showing an empirical evaluation of our wireless network, which is very important to fully understand the operating conditions for our implementations and tests. We continue by describing how we implemented our mobility management solutions; we do likewise for our QoS management solutions; in both cases detailed discussions are made followed by experiments that show the performance and results obtained by the imple-

mentations. It is worth noting that the implementation of our proposed in-band protocol —aiming for its integration with our mobility protocol— is still under development, so evaluating this integration is still pending and we can not currently show results. We are confident, however, that such integration constitutes only engineering work and that we have shown up to now sufficient proofs as to evidence the viability and usefulness of our approach. A resume of our contributions, the work that still needs to be done, and future approaches, are presented in Chapter 7.

Wireless Local Area Networks

Chapter 2

Wireless Local Area Networks

If you have built castles in the air, your work need not be lost; that is where they should be. Now put the foundations under them.

— Henry David Thoreau (1817 - 1862)

One important feature of IP networks is their flexibility to incorporate new technologies. Among them are wireless and mobile technologies. It is increasingly common, for instance, to access the Internet with a portable computer via a wireless local area network (WLAN). Among the different technologies available for wireless local networks, the most popular without a doubt is IEEE 802.11. Such popularity is evidenced by the number of products based on this standard that are commercially available. This fact, and also the fact that the platform we used for our prototypes and experiments is 802.11-based, explain why we often make reference to the standard throughout this document. Note should be taken though, that the architecture and mechanisms we discuss have a broader scope, *i.e.*, they are not restricted to the use of 802.11 access networks, but the proofs of concepts were made using this particular technology. As such, we will continue with a description of technologies used for access networks, making a particular emphasis on the IEEE 802.11 standard. Other technologies we will also discuss include Bluetooth and Hiperlan.

It is also worth noting that companies such as Airify[1] have announced products to support multiple wireless standards using the same network interface; this way, the same device could be used to take advantage of WLAN technologies such as 802.11 or Bluetooth, or wide area wireless such as GSM or GPRS. However, these type of products have been announced but have yet to be commercially available. Our proposals, being transparent to layer 2 technologies, could greatly benefit from these type of products.

Task Group	Activities
802.11	Initial standard, 2.4 GHz band, 2 Mbps
802.11a	High speed PHY layer in the 5 GHz band, up to 24 or 54 (optional) Mbps
802.11b	Higher speed PHY layer in the 2.4 GHz band, up to 11 Mbps
802.11d	New regulatory domains (countries)
802.11e	Medium Access Method (MAC) enhancements: multimedia, QoS, and enhanced security
802.11f	Inter-access-point protocol for AP interoperability
802.11g	Further higher data rate extension in the 2.4 GHz band, up to 22 Mbps
802.11h	Extensions for the 5 GHz band support in Europe

Table 2.1: Activities of the task groups working on the 802.11 standard

2.1 The IEEE 802.11 Standard

The Institute of Electrical and Electronic Engineers (IEEE) ratified the original 802.11 specification in 1997 as the standard for Wireless LANs (WLANs). That version of 802.11 provides for 1 Mbps and 2 Mbps data rates and a set of fundamental signaling methods and other services. The disadvantage with the original 802.11 standard are the slow data rates that are too slow to support most general business requirements. Recognizing the critical need to support higher data transmission rates, the IEEE ratified the 802.11b standard for transmissions of up to 11 Mbps. With 802.11b (also known as WiFi), WLANs are able to achieve wireless performance and throughput comparable to wired 10-Mbps Ethernet. 802.11a offers speeds of up to 54 Mbps, but runs in the 5 GHz band, so products based on this standard are not compatible with those based on 802.11b [107]. Currently, only WiFi-compatible products are available, but companies such as Atheros Communications[2] have announced products based on 802.11a for the end of this year (2001). Several Task Groups are working on further developments for the 802.11 standard, as shown in table 2.1.

Like all 802.x standards, 802.11 focuses on the bottom two layers of the OSI Reference Model: the Physical and the Data Link layers. In fact, the standard covers three physical layer implementations: direct-sequence (DS) spread spectrum, frequency-hopping (FH) spread spectrum, and infrared (IR). A single Medium-Access Control (MAC) layer supports all three physical layer implementations, as shown in figure 2.1. We will further discuss the two OSI layers that the 802.11 standard deals with.

[1] `http://www.airify.com`
[2] `http://www.atheros.com`

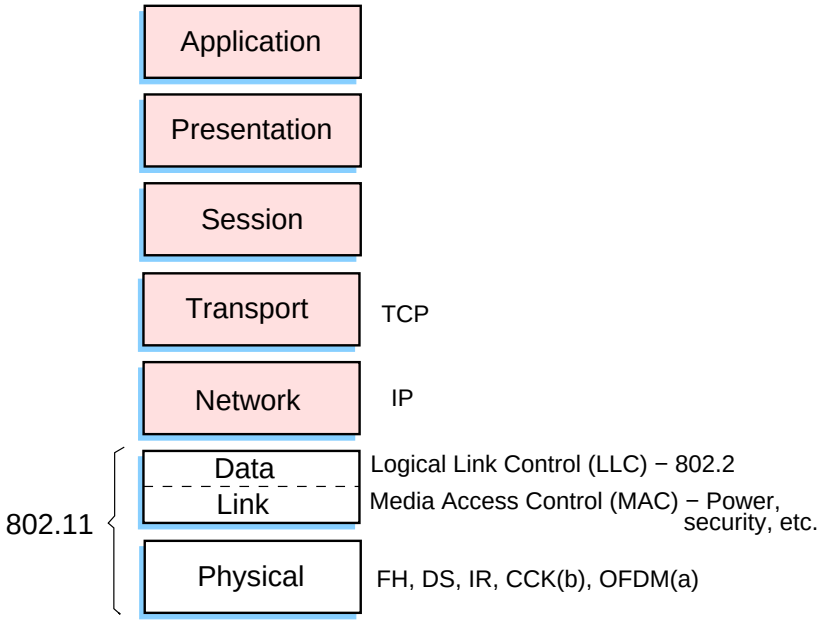

Figure 2.1: The 802.11 standard and the ISO model

2.1.1 802.11 Architecture

Each computer, mobile, portable or fixed, is referred to as a station in 802.11. Mobile stations access the LAN during movement. The 802.11 standard defines two modes: *infrastructure* mode and *ad hoc* mode. In infrastructure mode (figure 2.2), the wireless network consists of at least one access point (AP) connected to the wired network infrastructure and a set of wireless end stations. This configuration is called a *Basic Service Set (BSS)*. An *Extended Service Set (ESS)* is a set of two or more BSSs forming a single subnetwork. Two or more DSSs are interconnected using a Distribution System (DS). In an ESS the entire network looks like an independent BSS to the Logical Link Control (LLC) layer; this means that stations within the ESS can communicate or even move between BSSs transparently to the LLC. The DS can be thought of as a backbone network that is responsible for MAC-level transport of MAC service data units (MSDUs). The DS, as specified by IEEE 802.11, is implementation-independent. Therefore, the DS could be a wired IEEE 802.3 Ethernet LAN, IEEE 802.4 token bus LAN, IEEE 802.5 token ring LAN, fiber distributed data interface (FDDI) metropolitan area network (MAN), or another IEEE 802.11 wireless medium. Note that while the DS could physically be the same transmission medium as the BSS, they are logically different, because the DS is solely used as a transport backbone to transfer packets between different BSSs in the ESS. An ESS can also provide gateway access for wireless users into a wired network such as the Internet. This is accomplished via a device know as a *portal*. The portal is a logical entity that specifies the integration point on the DS where the IEEE 802.11 network integrates with a non-IEEE 802.11 network. If the network is an IEEE 802.x, the portal incorporates functions that are analogous to a bridge; that is, it provides range extension and the translation between different frame formats.

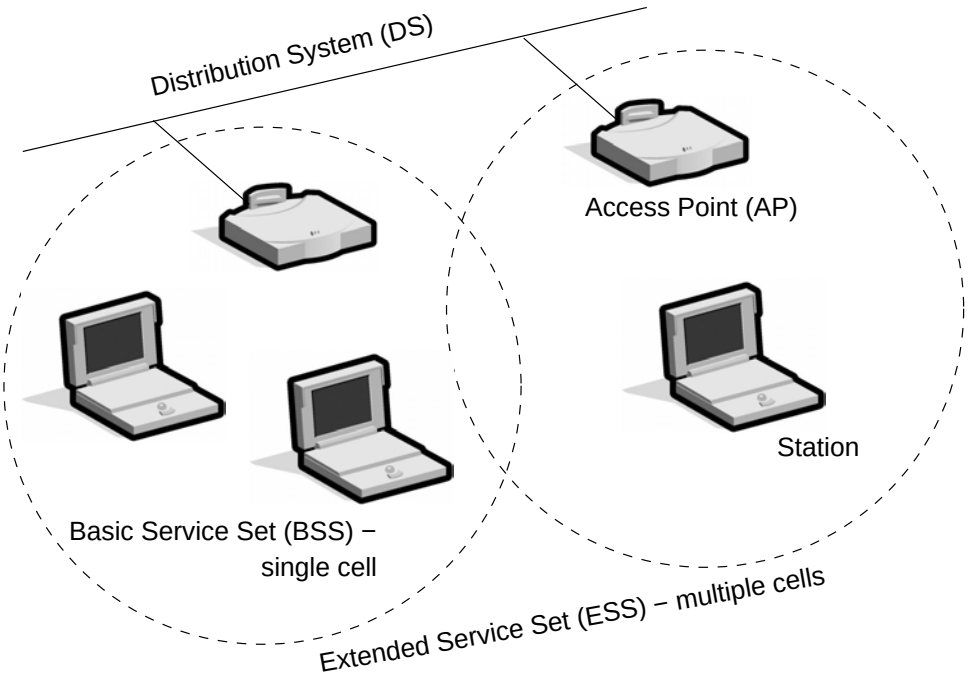

Figure 2.2: Infrastructure Mode.

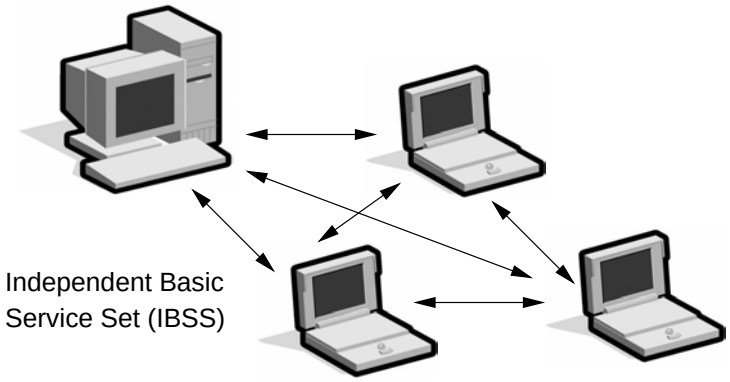

Figure 2.3: Ad Hoc Mode.

Data Rate	Code Length	Modulation	Symbol Rate	Bits/Symbol
1 Mbps	11 (Barker Sequence)	BPSK	1 MSps	1
2 Mbps	11 (Barker Sequence)	QPSK	1 MSps	2
5.5 Mbps	8 (CCK)	QPSK	1.375 MSps	4
11 Mbps	8 (CCK)	QPSK	1.375 MSps	8

Table 2.2: Technical characteristics of the 802.11 standard at different rates

The ad hoc mode (also called peer-to-peer mode or an Independent Basic Service Set, or IBSS) is simply a set of 802.11 stations that communicate directly with one another without using an access point or any connection to a wired network (figure 2.3). In ad hoc networks there is no base and no one gives permission to talk; these networks are spontaneous and can be set up rapidly, but are also limited both temporally and spatially.

2.1.2 The Physical Layer

The three physical layers originally defined in the 802.11 included two spread-spectrum radio techniques and a diffuse infrared specification. The radio-based standards operate within the 2.4 GHz ISM band. These frequency bands are recognized by international regulatory agencies, such as the FCC (USA), ETSI (Europe) and the MKK (Japan) for unlicensed radio operations. As such, 802.11-based products do not require user licensing or special training. Spread-spectrum techniques, in addition to satisfying regulatory requirements, boost throughput, and allow many unrelated products to share the spectrum without explicit cooperation and with minimal interference.

The original 802.11 wireless standard defines data rates of 1 Mbps and 2 Mbps via radio waves using frequency hopping (FH) spread spectrum or direct-sequence (DS) spread spectrum. It is important to note that FH and DS are fundamentally different transmission mechanisms and will not interoperate with each other.

Kamerman [75] has compared various performance aspects of the 802.11 standard's DS and FS spread spectrum techniques. DS has a more robust modulation and a larger coverage range than FH, even when FH uses twice the transmitter power output level. FH gives a large number of hop frequencies, but the adjacent channel interference behavior limits the number of independently operating collocated systems. Hop time and a smaller packet size introduce more transmission time overhead into FH, which affects the maximum throughput. Although FH is less robust, it gives a more graceful degradation in throughput and connectivity. Under poor channel and interference conditions, FH will continue to work over a few hop channels a little longer than over the other hop channels. DS, however, still gives reliable links for a distance at which very few FH hop channels still work. For collocated networks (access points) DS gives a higher potential throughput with fewer access points than FH, which has more access points. The smaller number of access points used by DS lowers the infrastructure cost.

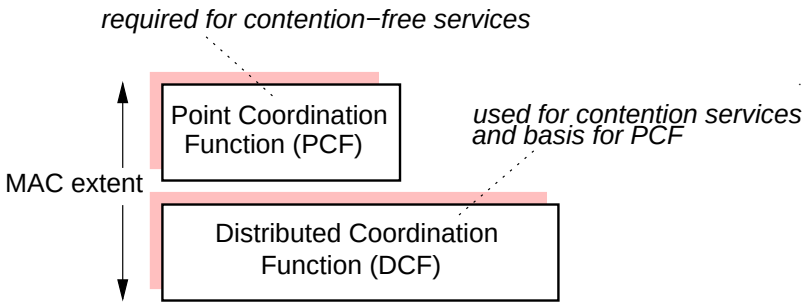

Figure 2.4: 802.11 MAC architecture.

2.1.3 The Data Link Layer

The Data Link layer within 802.11 consists of two sublayers: Logical Link Control (LLC) and Media Access Control (MAC). 802.11 uses the same 802.2 LLC and 48-bit addressing as other 802 LANs, allowing for very simple bridging from wireless to IEEE wired networks, but the MAC is unique to WLANs.

Of particular interest in the specification is the support for two fundamentally different MAC schemes to transport asynchronous and time-bounded services. The first scheme, distributed coordination function (DCF), is similar to traditional legacy packet networks supporting best-effort delivery of the data. The DCF is designed for asynchronous data transport, where all users with data to transmit have an equally fair chance of accessing the network. The point coordination function (PCF) is the second MAC scheme. The PCF is based on polling that is controlled by an access point (AP). The PCF is primarily designed for the transmission of delay-sensitive traffic. As can be seen in Figure 2.4, the DCF is the basis for the optional PCF.

The basic access method, DCF, is drawn from the family of carrier-sense with collision avoidance (CSMA/CA) protocols. The collision detection (CD) mechanism —as used in the CSMA/CD protocol of Ethernet— can not be used under 802.11 due to the *near/far problem*: to detect a collision, a station must be able to transmit and listen at the same time, but in radio systems the transmission drowns out the ability of the station to hear a collision. So, 802.11 uses CSMA/CA under which collisions are avoided by using explicit packet acknowledgment (ACK) to confirm that the data packet arrived intact.

The basic principles of CSMA/CA are *listen before talk* and *contention*. The protocol starts by listening on the channel (the Carrier Sense part), and if it is idle, it sends the first packet in the transmit queue (after a short time, as will be explained later). If it is busy (either another transmission or interference), the station waits for the end of the current transmission and then starts the contention (wait a random amount of time). When its contention timer expires, if the channel is still idle, the station sends the packet. The station having chosen the shortest contention delay wins and transmits its packet. The other stations just wait for the next contention (at the end of this packet). Because the contention is a random number and done for every packet, each station is given an equal chance to access the channel (on average, since this is an statistic method).

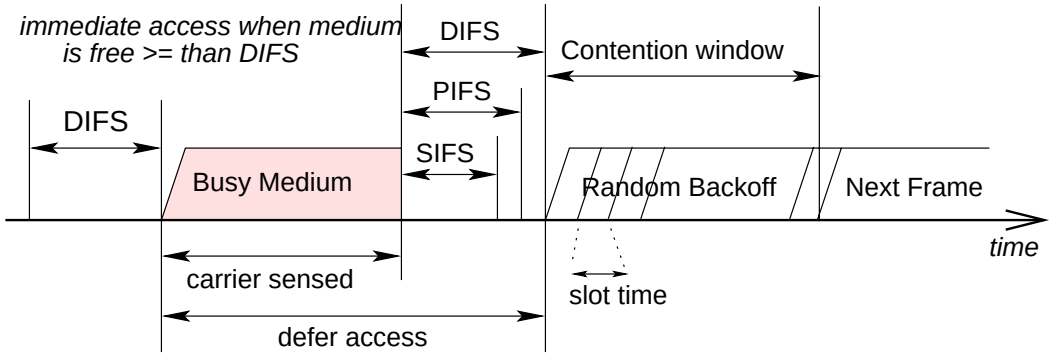

Figure 2.5: Primary access mechanism.

As previously stated, collisions can not be detected by the radio systems, and since the radio needs time to switch from receive to transmit, the contention is usually slotted – a transmission may start at the beginning of a slot: 50 μs in 802.11 FH and 20 μs in 802.11 DS. This makes the average contention delay larger, but significantly reduces the number of collisions. The random backoff time is distributed according to a uniform distribution (in discrete slot times) where the maximum extent of the uniform range is called the *contention window* (CW). The CW parameter, that is, the range of this uniform distribution, is doubled each time a frame transmission is unsuccessful, as determined by the absence of an acknowledgment (ACK) frame. The backoff time is thus determined as $Backoff\_time = \lfloor 2^{2+i} \times rand() \rfloor \times slot\_time$ where i is the number of consecutive times a station attempts to transmit, $rand()$ is an uniform random variate in (0,1) and $\lfloor x \rfloor$ represents the largest integer less than or equal to x.

The exponential backoff mechanism helps reduce collisions in response to increasing numbers of contending stations. Furthermore, as shown in figure 2.5, there is an initial interframe space (IFS) that can take on three different values representing priorities for transmission. the highest-priority frames are transmitted using the short IFS (SIFS). For example, the immediate acknowledgment that a receiving station sends back to the transmitting station makes use of the SIFS to guarantee that no other station intervenes. The next longest IFS, the point coordination function IFS (PIFS), is used to provide a priority mechanism by which time-critical frames can be transmitted before asynchronous data frames, which use the longest IFS, the distributed coordination function IFS (DIFS).

The *hidden node* issue is a MAC layer problem specific to wireless in which two stations on opposite sides of an access point can both hear activity from an access point, but not from each other, usually due to distance or an obstruction. The hidden node problem is exemplified in figure 2.6, where node C can not hear node A; so, if node A is transmitting, node C will not know and may transmit as well, which will generate collisions. To solve this problem, 802.11 specifies an optional Request to Send/Clear to Send (RTS/CTS) protocol at the MAC layer. When this feature is in use, a sending station transmits an RTS and waits for the access point to reply with a CTS – this waiting time is known as the *network allocation vector* (NAV). Since all stations in the network can hear the access point, the CTS causes them to delay any intended transmission, allowing the sending station to transmit and receive a packet acknowledgment without any chance of collision.

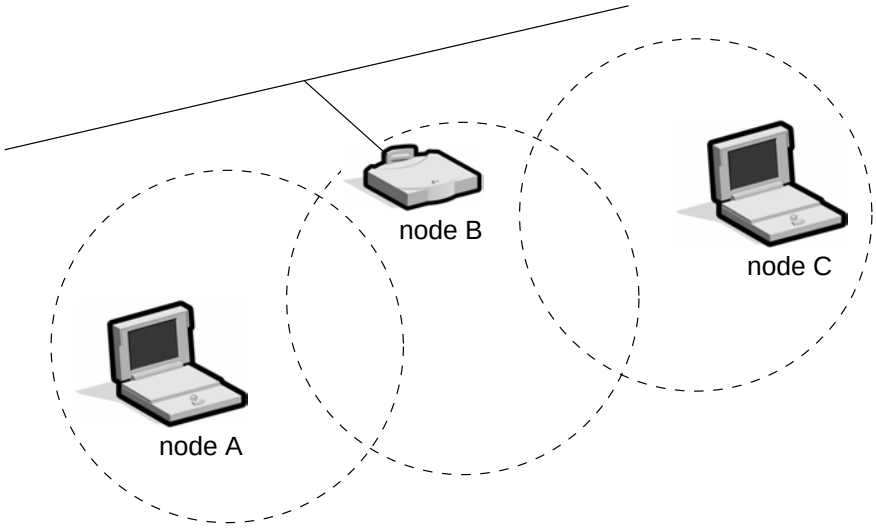

Figure 2.6: The hidden node problem in wireless networks.

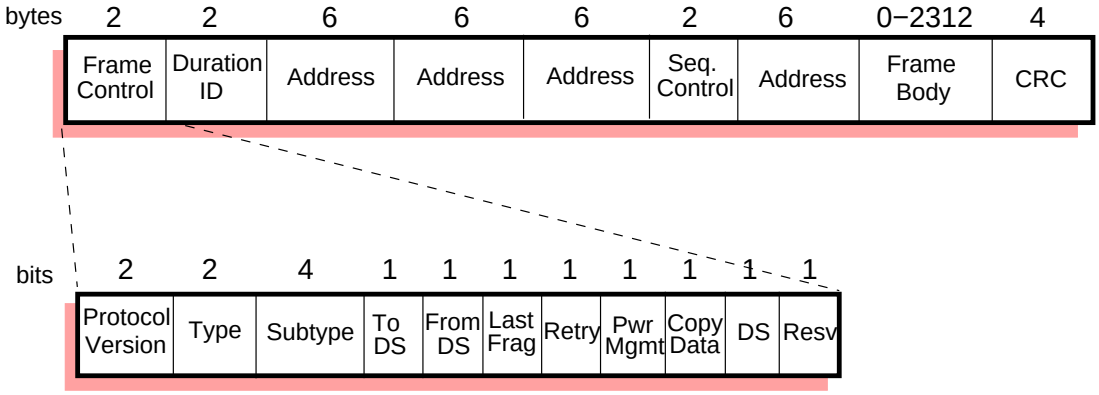

Figure 2.7: Standard IEEE 802.11 frame format.

Since RTS/CTS adds additional overhead to the network by temporarily reserving the medium, it is typically used only on the largest-sized packets, for which retransmission would be expensive from a bandwidth standpoint.

IEEE 802.11 supports three different types of frames: management, control and data. The management frames are used for station association and disassociation with the AP, timing and synchronization, and authentication and deauthentication. Control frames are used for handshaking during a Contention Period (CP), for positive acknowledgment during the CP, and to end the Contention-Free Period (CFP). Data frames are used for the transmission of data during the CP and CFP, and can be combined with polling and acknowledgments during the CFP. Figure 2.7 shows the standard IEEE 802.11 frame format. Note that the frame body (MSDU) is a variable-length field consisting of the data payload and 7 octets for encryption/decryption if the Wired Equivalent Privacy (WEP) protocol is implemented. The IEEE standard 48-bit MAC addressing is used to identify

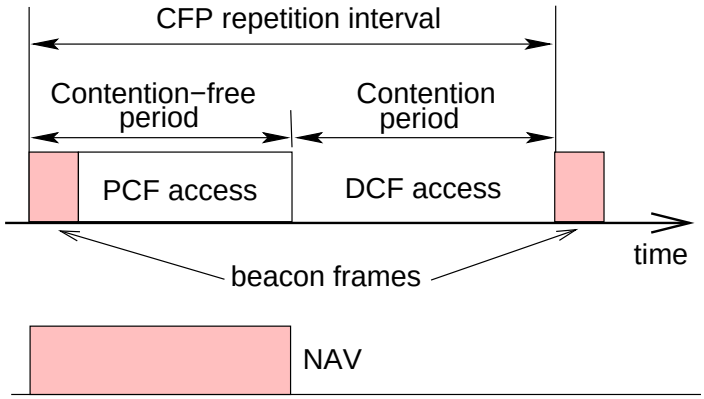

Figure 2.8: Alternation of contention-free and contention periods.

a station. The 2 duration octets indicate the time (in microseconds) the channel will be allocated for successful transmission of a MAC protocol data unit (MPDU). The type bits identify the frame as either control, management, or data. The subtype bits further identify the type of frame (*e.g.*, CTS control frame). A 32-bit cyclic redundancy check (CRC) is used for error detection.

In order to support time-bounded services, the 802.11 standard specifies the optional use of the aforementioned point coordination function (PCF) in which a point coordinator (or PCF station[3]) has a high priority control of the medium. That is, when the PCF is active, the PCF station allows only a *single* station in each cell to have priority access to the medium at any time. This is implemented through the use of the previously mentioned PIFS and a beacon frame (Fig. 2.8) that notifies all[4] of the other stations in the cell not to initiate transmissions for the length of the contention-free period (CFP). Having silenced all the stations, the PCF station can then allow a given station to have contention-free access through the use of an (optional) polling frame that is sent by the PCF station. Note that the length of the CFP can vary within each CFP repetition interval according to the system load. A typical wireless LAN installation would use different channels for adjacent cells to prevent two PCF stations (*i.e.*, access points) from using (and hence colliding on) the same channel during the CFP. This would allow coexistence, even on the same channel, with an ad hoc network that is using DCF only.

2.2 Other Related Standards

There are other WLAN technologies available besides 802.11, and we will review some of the most prominent ones, namely Bluetooth and Hiperlan. It is relevant to point out that up to now (Q4 2001) the market for WLANs is dominated by products based on the 802.11

[3]The PCF station is always an access point, so the use of the PCF and hence support for time-bounded services is limited to networks with infrastructure

[4]If one of the stations does not hear the expected beacon, it sets its NAV to a known maximum value for the length of the CFP

standard. There are also starting to appear some products based on Bluetooth but they have been very deceiving and many important equipment and software manufacturers have decided not to support this standard [137; 109], at least temporarily. Although some early prototypes for HiperLAN 2 have been demonstrated [46], there are not any commercial products available yet.

2.2.1 HiperLAN

Between 1990 and 1992, the European Telecommunications Standards Institute (ETSI) noticed the trend towards faster and better wireless networks and started the development of standards for this type of networks. Within this framework, the Broadband Radio Access Networks (BRAN) project[5] of ETSI is working on a standard called High Performance Radio Local Area Network (HiperLAN). This project quickly separated intro four different HiperLAN types:

- *HiperLAN 1.* A standard for *ad-hoc* networking operating in the 5.2 GHz band with a spectrum of 100 MHz and with speeds of up to 19 Mbps. It offers one-to-one communications as well as one-to-many broadcasts. Using the CSMA/CA technique for resolving contention, the scheme shares available radio capacity between active users who attempt to transmit data during an overlapping time span. Although HiperLAN 1 provides a means of transporting time-bounded services, it does not control nor guarantees QoS on the wireless link. This is what motivated ETSI to develop a new generation of standards that support asynchronous data and time-critical services that are bounded by specific time delays.

- *HiperLAN 2.* Specifies a radio-access network that can be used with a variety of core networks (*e.g.*, IP, ATM, UMTS). HiperLAN 2 also operates in the 5.2 GHz band with 100 MHz spectrum, but at speeds of up to 54 Mbps [82]. We will give further details on this standard ahead. Since it is, up to now, the standard in the HiperLAN family that is best suited for our work.

- *HiperAccess.* This is the next step from HiperLAN 2 providing outdoor wireless access. It gives up to 5 Km coverage between wireless access points and wireless termination points and is therefore intended for stationary and semi-stationary applications. The original operating frequencies were in the 5 GHz band, but this is currently under discussion.

- *HiperLink.* The standard is meant to provide interconnecting service for high data rate sources such as networks (*e.g.*, HiperLANs). Therefore, HiperLink provides point-to-point interconnections at very high data rates of up to 155 Mbps over distances up to 150 meters. The operating frequency is in the 17 GHz band with 200 MHz spectrum at the moment.

The standard for HiperLAN 1 was finalized during 1996, although amendments to it were made during 1998. HiperLAN types 2-4 were designed to support only ATM

[5] `http://www.etsi.org/bran/`

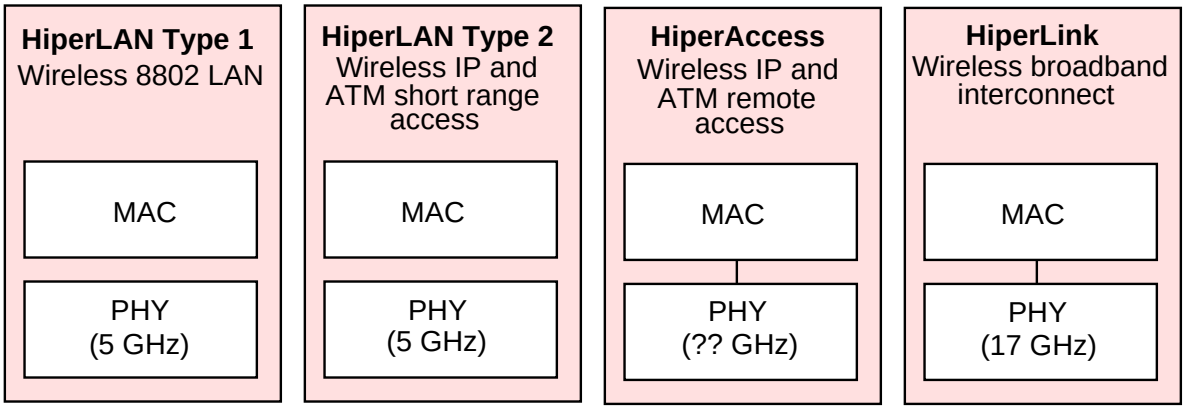

Figure 2.9: Overview of HiperLAN standards.

networks, but at the moment HiperLAN 2 also supports access to IP and UMTS networks. The names for types 3 and 4 were changed to HiperAccess and HiperLink, respectively. Figure 2.9 gives an overview of the different HiperLAN standards.

HiperLAN 2

The standard specifies a radio access network that can be used with a variety of core networks. Since 1999, the HiperLAN 2 Global Forum[6] promotes the adoption of HiperLAN 2 [74]. The following are some of the general features of HiperLAN 2:

- *High-speed transmission.* Transmission rates go up to 54 Mbps on the physical layer and up to 25 Mbps on layer 3. In order to achieve this, a modulation method called Orthogonal Frequency Digital Multiplexing (OFDM) is used. Above the physical layer, the MAC protocol is all new which implements a form of dynamic time-division duplex to allow for a highly efficient utilization of radio resources.

- *Connection-oriented.* Connections are established between the mobile terminals (MTs) and the access points (APs) prior to the transmission of data, using signaling functions on the control plane. Connections are time-division multiplexed over the air interface.

- *QoS support.* The connection-oriented nature of HiperLAN 2 makes it straightforward to implement support for QoS. Each connection can be assigned a specific QoS, for instance in terms of bandwidth, delay, jitter, bit error rate, etc. It is also possible to use a more simplistic approach where each connection can be assigned a priority level relative to other connections.

- *Automatic frequency allocation.* Access points have built-in support for automatically selecting an appropriate radio channel for transmission within each AP's coverage area. An AP listens to neighboring APs as well as to other radio sources in the environment and selects an appropriate radio channel.

[6] `http://www.hiperlan2.com`

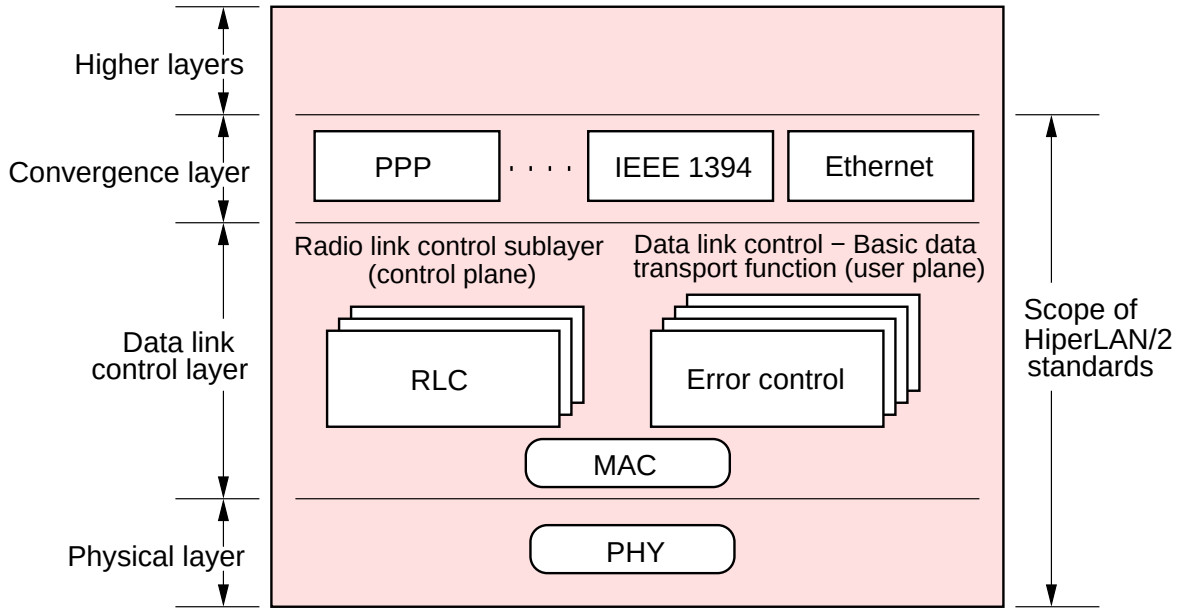

Figure 2.10: Architecture of the HiperLAN 2 protocols.

- *Mobility support.* A MT uses the AP with the best radio signal as measured by the signal-to-noise ratio. Thus, as the MT moves around, it may perceive better reception from an alternative AP and decide to perform a handover[7] to it. During handover, some packet loss may occur.

Figure 2.10 depicts the architecture of the HiperLAN 2 protocols, which is very flexible since it defines core network independent physical (PHY) and data link control (DLC) layers, and a set of convergence layers have been or are currently being defined for interworking with IP, ATM, third-generation core networks, and networks that use IEEE 1394 (Firewire) protocols and applications.

2.2.2 Bluetooth

Bluetooth is a protocol intended to wirelessly connect cellular phones, laptops, handheld computers, digital cameras, printers, and other devices [57]. It operates over short distances of up to 10 meters, basically being a wireless replacement for data cables and infrared connections. There are currently some discussions going on in order to extend its range to 100 meters by increasing the transmit power to 100 mW. Although Bluetooth was initially developed by Ericsson in the late 1990s, it is currently led by the Bluetooth SIG[8] including members such as Nokia, IBM, Toshiba, Intel, 3Com, Motorola, Lucent Technologies and Microsoft. It is then not a technology backed by an standards body, but instead backed by an industry consortium.

[7] The terms handover and handoff are used interchangeably in the literature.

[8] `http://www.bluetooth.com`

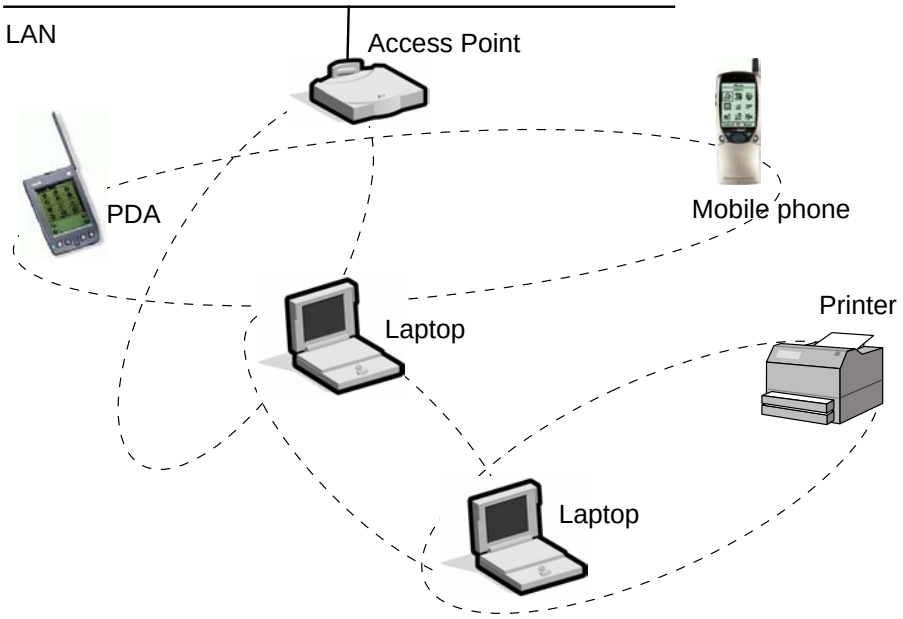

Figure 2.11: A Bluetooth scatternet of four piconets.

The Bluetooth system supports both point-to-point or a point-to-multipoint connections. In point-to-multipoint, the channel is shared among several bluetooth units. Two or more units sharing the same channel form a *piconet*. There is one master unit and up to seven active slave units in a piconet. These devices can be in either of the following states: active, park, hold and sniff. Multiple piconets with overlapping coverage areas form a *scatternet* (figure 2.11).

The Bluetooth system consists of a radio unit, a link control unit and a support unit for link management and host terminal interface functions. The radio operates in the 2.4 GHz ISM (Industry, Science and Medicine) band. Depending on the class of the device, a bluetooth radio can transmit up to 100 mW (20 dBm) to minimum of 1 mW (0 dBm) of power. It uses frequency hopping for low interference and fading, and a TDD (Time-Division Duplex) scheme for full duplex transmission and transmits using GFSK (Gaussian Frequency Shift Keying) modulation [58].

The Bluetooth protocol uses a combination of circuit and packet switching. The channel is slotted and slots can be reserved for synchronous packets. The protocol stack can support an asynchronous connectionless link (ACL) for data and up to three simultaneous synchronous connection-oriented (SCO) links for voice or a combination of asynchronous data and synchronous voice (DV packet type). Each voice channel supports a 64 Kb/s synchronous channel in each direction. The asynchronous channel can support maximum of 723.2 Kb/s uplink and 57.6 Kb/s downlink (or viceversa) or 433.9 Kb/s symmetric links. The stack (shown in figure 2.12) primarily contains a physical level protocol (Baseband) and a link level protocol (LMP) with an adaptation layer (L2CAP) for upper layer protocols to interact with lower layer ones.

The current Bluetooth specification can support QoS demanding applications by either

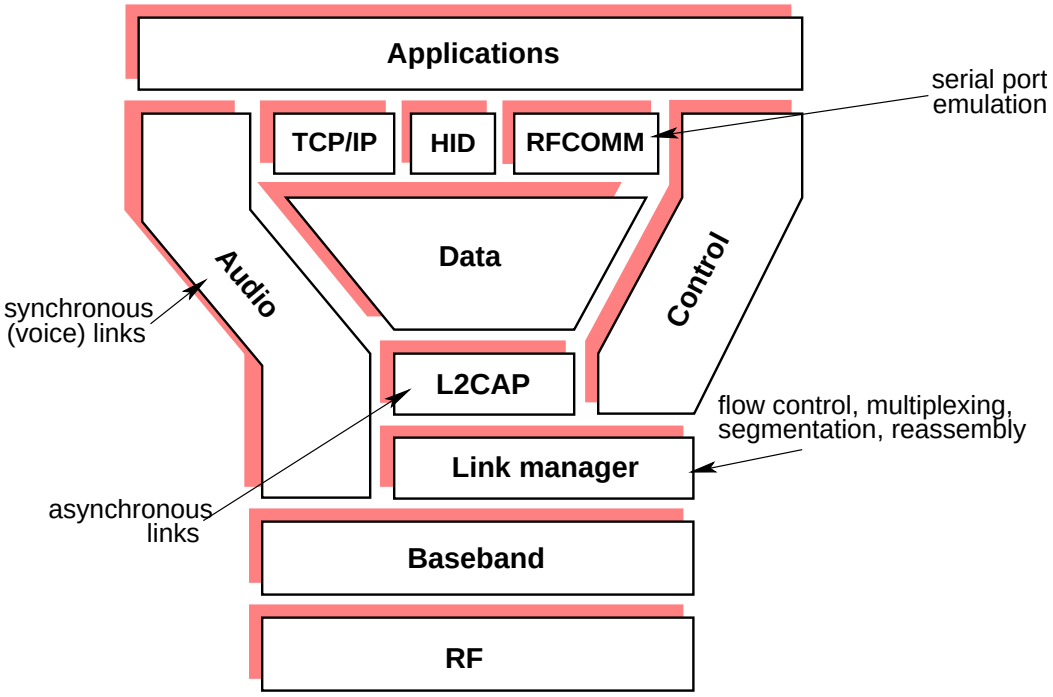

Figure 2.12: The Bluetooth protocol stack

using the SCO link of the Guaranteed class in the ACL link. Van der Zee [42] has presented a Bluetooth QoS framework that provides additional QoS support for Bluetooth on top of the ACL link. These additional QoS classes are: Priority, Isochronous and Low Bit Rate Low Delay (LBLD). Bluetooth technologies support the following QoS parameters: bandwidth, delay, delay variation, reliability, and ordering.

It should be clear, given its distance coverage, bandwidth and other characteristics, that Bluetooth does not really fit within the profile for supporting WLANs —as their promoters pretend through intense marketing campaigns. Bluetooth fits more within the profile of technologies used for Wireless Personal Area Networks (WPANs), as those studied by the IEEE 802.15 Working Group[9]. However, given its support for local wireless connectivity, it is relevant for our work in micro-mobility and that is why we have considered to introduce it here.

2.3 Chapter conclusions

Given their flexibility, ease of installation, dropping prices, and increasingly higher speeds, WLANs are gradually replacing many wired LANs as the networks of choice for typical activities such as Internet access. There are currently co-existence problems between some of the technologies we mention here, namely between 802.11 and Bluetooth. A source of problems is the fact that Bluetooth has been designed to transmit blindly, whenever its

[9] `http://grouper.ieee.org/groups/802/15/`

timing dictate, as if there was no possibility that a collocated system might be using the same frequency (as 802.11 does). This has earned it the reputation of a "bad neighbor" in the 2.4 GHz band. There are also other common sources of interference in this unregulated band, including microwave ovens and newer generations of cordless phones. Historically, microwave ovens are by far the most significant source of interference in residential and office environments [76], but with the impending avalanche of new communication devices with embedded Bluetooth radios, serious questions have been raised about their interference on wireless LANs. The IEEE 802.15 WPAN Task Group 2 (TG2) [10] is developing recommended practices and mechanisms to facilitate the coexistence between WLANs (such as 802.11) and WPANs (such as Bluetooth).

Even if its the most widely used WLAN technology, 802.11 does not offer the best support for QoS. Contrary to 802.11, Bluetooth and Hiperlan have been designed from the beginning with QoS support features. The Point Coordination Function (PCF) of 802.11 that allows access to the medium to only one station per cell can not be considered a QoS feature. The 802.11e working group is currently designing a variation of the basic 802.11 protocol to incorporate QoS support, but that is still work in progress. The CSMA/CA access method used in 802.11 poses serious problems for supporting QoS as important overhead is introduced by backoff times and waiting periods (*e.g.*, SIFS, DIFS) where stations are not allowed to send traffic into the medium. In fact, this access method is designed to allow fair access to all stations, and not the priority access necessary for QoS. Further problems arise due to factors such as distance between the source and the destination, signal interference, and fading.

[10] `http://grouper.ieee.org/groups/802/15/pub/TG2.html`

Quality of Service in the Context of the Internet Protocols

Chapter 3

Quality of Service in the Context of the Internet Protocols

Democracy does not guarantee equality of conditions - it only guarantees equality of opportunity.

— *Irving Kristol*

3.1 Quality of Service overview

Quality of Service (QoS) is a concept that exists even in non-computerized systems for transporting information. The postal system offer services for standard, express, and over-night delivery. Obviously, clients will pay different prices according to the quality of the service. The same concept of granting different levels of service to users can be applied to computer networks, since not all applications have the same requirements and thus, in order to make an efficient use of available resources, they should obtain a quality of service according to their needs.

The notion of QoS was not common (or even non-existent) at the time when the Internet protocols first started being developed. For this reason, service quality on IP networks has always taken a best-effort approach, where no hard guarantees can be given regarding the provision of a certain service level. In that best effort model, an application will receive whatever level of performance (*e.g.*, end-to-end packet delay and loss) that the network is able to provide at that moment. More recently, the IETF has proposed different architectures to provide QoS on the Internet, namely the Integrated Services (IntServ) and the Differentiated Services (DiffServ) models.

3.1.1 Defining quality of service

QoS: a short acronym with an apparently simple definition that, however, despite several years of research within the networking community, still means different things to different people. The intuitive and most generalized definition says that the QoS concept deals with the system characteristics that have an influence over the quality perceived by the application [51].

There are also some definitions given by standards organisms such as ITU-T [86] — which refers to QoS as "a set of requirements about the collective behavior of one or more objects"— that are somewhat vague and fuzzy. Also, as Black [17] notes, the term quality of service was used some twenty years ago in the OSI model, and its meaning refers to the ability of a service provider to support user's application requirements with regard to at least four service categories: a) bandwidth, b) latency (delay), c) jitter, and d) traffic loss.

Most definitions commonly state that applications specify their QoS requirements, while the system provides (or tries to provide) QoS guarantees. In fact, a lot of things can happen between these two actions, since the system must first determine if it has enough resources to satisfy the requirements; if it does, it will reserve the necessary resources, if not, it will either deny the application's request or possibly suggest a set of requirements that will be able to satisfy. In this latter option, the application could accept the suggestion and continue its execution, or choose not to accept it and be subject to be rejected by the system. The application could, of course, later retry a negotiation hoping that the system has freed resources. Thus, the following elements are commonly involved in order to provide QoS guarantees:

- A QoS *specification* mechanism by which applications may specify their requirements.
- *Admission control* to determine if new applications can be admitted into the system without affecting the QoS level of others already admitted.
- A system for *negotiation* (and re-negotiation) of QoS to serve as many applications as possible.
- *Reservation and scheduling* of resources to satisfy the requirements of admitted applications.
- *Monitoring*, such as traffic policies, to assure that applications do not exceed their accepted requirements.

Thus, QoS provision in a networked environment requires the use of several mechanisms —like the ones stated above— in network nodes. Also, a common agreement is that it is necessary to differentiate between traffic or service types so one or more classes of traffic can be treated differently than other types. Indeed, in order to provide QoS, it is necessary to introduce unfairness: for instance, the capability to provide premium treatment for one class of traffic and only best effort with no guarantees for another class [48]. To better understand the concept of QoS and how it is provided, in the next subsection we will discuss the involved mechanisms.

3.1.2 QoS-enabling mechanisms

It has been noted [48] that, unless the network is over-engineered to accommodate all possible users with no congestion in any portion of the network, strict adherence to the fair allocation of per-user bandwidth is not a very plausible scheme. Thus, as already stated, unfairness introduced by traffic differentiation is used in the provision of QoS. In the common case of networks where traffic demand exceeds the offer of resources (*i.e.*, bandwidth), the exceeding traffic starts to accumulate in router's queues, generating congestion, and some mechanisms must be set-up to handle the situation. Any network that admits traffic and users on a demand basis (statistical multiplexing) must deal with the problem of congestion. The end result of congestion is the reduction in traffic throughput and increased delays in the delivery of the traffic to the receiver. Congestion control approaches can be classified under two categories, based on whether or not the transport layer receives any explicit assistance from the network layer in order to control the congestion that has built up [87]:

- *End-end approaches.* The network layer provides no explicit support to the transport layer for congestion control purposes, so even the mere presence of congestion in the network must be inferred by the end systems, based on observed network behavior (*e.g.,*, packet loss and delay). A typical example is the case of TCP, since the IP layer does not provide any feedback regarding network congestion. Using TCP, a source increases its window size until it detects a packet loss. At this point, the source reduces the window size, and the cycle repeats. Two phases are distinguished in the process: an exponential phase (slow-start) and a linear phase. Some variants of TCP have been proposed, of which TCP Tahoe and TCP Reno are two of the most widely used ones. TCP Tahoe detects losses using timeouts. TCP Reno detects losses using both timeouts and the receipt of three ACKs with the same cumulative sequence number, entering then a procedure called fast recovery. Keshav [81] discusses different variations of TCP with great detail.

- *Network-assisted approaches.* Network-layer components (*i.e.*, routers) provide explicit feedback to the sender regarding the congestion state in the network. This feedback may be as simple as a single bit indicating congestion at a link (as used in the IBM SNA [131], and DECnet [71] architectures, and proposed for TCP/IP networks [125]). More sophisticated network-assisted approaches are also possible, as is the case with one form of ATM ABR congestion control in which the switch can inform the sender of a transmission the rate on an outgoing link. Network-assisted congestion control is depicted in Figure 3.1: congestion information is typically fed back to the network in one of two ways. First, direct feedback may be sent from a network router to the sender, typically in the form of a choke packet (essentially saying "I'm congested!"). Second, a router along the way from sender to receiver marks/updates a field in a traversing packet to indicate congestion. This way, upon receipt of the packet, the receiver may be able to inform the sender that congestion is present. Obviously, this type of notification may take at least one full round-trip time.

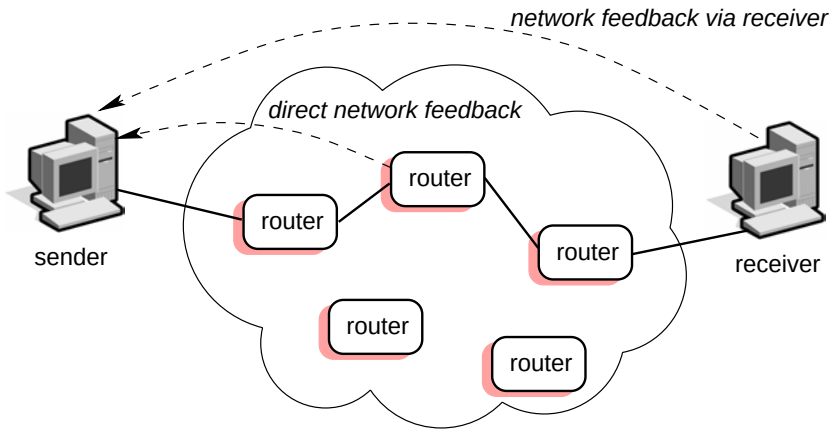

Figure 3.1: Network-assisted approach for congestion control.

It should be emphasized that flow control is often confused with congestion control [81]. Congestion refers to a sustained overload of intermediate network elements. Flow control refers to the set of techniques that enable a data source to match its transmission rate to the currently available service rate at a receiver and in the network. Thus, flow control is one mechanism for congestion control.

When congestion occurs and queues start forming, **scheduling** is carried out by using a queuing algorithm to decide in what order service requests (incoming packets) are allowed to resources (output queues and output lines), and also to manage the service queues (output buffers). There is a great number of queuing algorithms, but only work-conserving scheduling disciplines [84] —which means that the router is never idle when there are packets to be serviced— are widely used. Non-work-conserving disciplines are considered more of a research issue because systems based on them are complex to build, and are not widely used [15]. Some of the most popular queuing algorithms in use are:

- *FIFO.* The First-In First-Out (FIFO), also known as First-Come First-Served (FCFS), is one of the simplest algorithms, since it involves buffering and forwarding of packets in the order of arrival. FIFO embodies no concept of priority or traffic classes and consequently makes no decision about packets priority. There is only one queue and packets are treated equally. When FIFO is used, ill-behaved sources can consume all the bandwidth, bursty sources can cause delays in time-sensitive or important traffic, and important traffic can be dropped because less important traffic fills in the queue.

- *Priority Queuing.* With Priority Queuing (PQ), packets are identified and given a certain priority, then those belonging to one priority class of traffic are sent before all lower priority traffic to ensure timely delivery of those packets. PQ could certainly be considered a primitive form of traffic differentiation, but the approach is less than optimal. A low priority queue can be detrimentally affected, and, in the worst case, never allowed to send its packets if a limited amount of bandwidth is available or if the transmission rate of critical traffic is high (causing starvation).

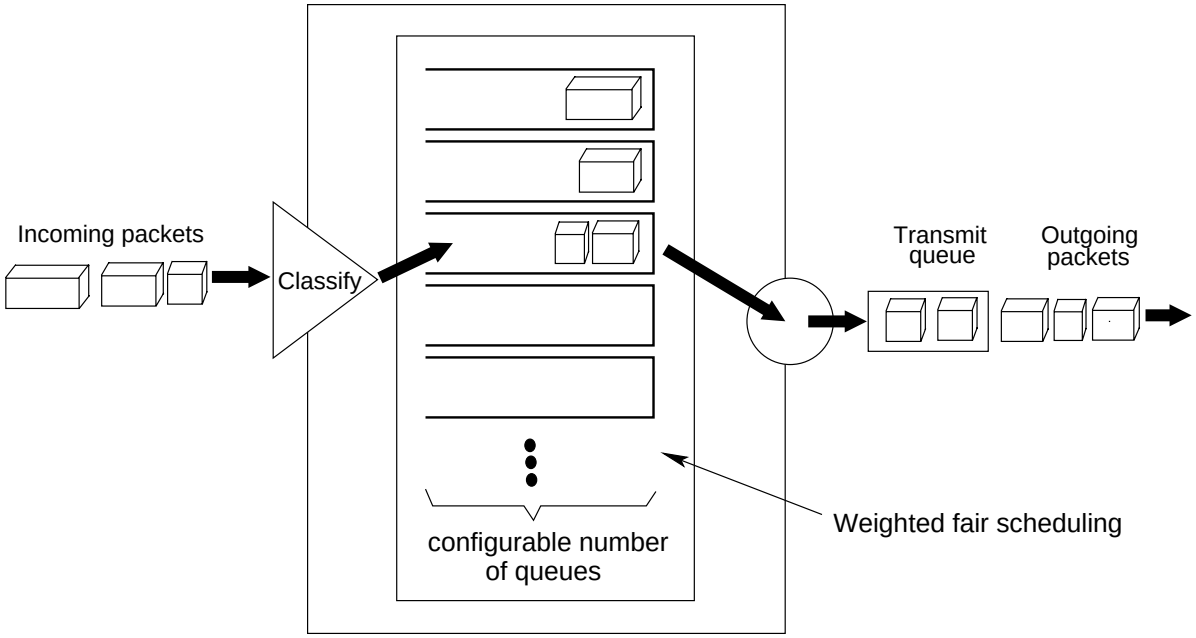

Figure 3.2: Weighted Fair Queuing (WFQ) operation.

- *WFQ*. The Weighted Fair Queuing (WFQ) service discipline, part of the original work on Generalized Processor Sharing (GPS) [110][1], and its many variants have been very popular due, in part, to the fact that it overcomes some of the limitations of the FIFO and PQ schedulers by allowing for a fine-grain control over the service received by individual flows. WFQ uses a servicing algorithm that attempts to provide predictable response times and negate inconsistent packet transmission timing; it does this by sorting and interleaving individual packets by flow and queuing each flow based on the volume of traffic in this flow (see Figure 3.2). Using this approach, larger flows are prevented from consuming network resources (*e.g.*, bandwidth), which could eventually starve smaller flows; this is the fairness aspect of the WFQ. There are several variations of WFQ [141; 135] of which Worst-case Fair Weighted Fair Queuing (WF^2Q) [10] is an usual one.

Scheduling plays an important role in the QoS provided by the network, but it is also necessary to have sufficient buffers to hold incoming packets. Given currently available high link speeds, the amount of memory required to buffer traffic during transient periods of congestion can be large and exceed the amount of memory that most routers and switches have. Packets that arrive during these transient periods will have to be dropped. As Guerin indicates [54], in order to avoid a haphazard behavior when a link experiences congestion, several different **buffer (queue) management** schemes can be used. These can be roughly classified along the following two dimensions:

1. When are packet discard decisions made? Typically, packet discard decisions are

[1] According to Partridge [113], the concept of Fair Queuing (FQ) was developed by Nagle [103] and later refined by Demers *et al.* [41].

made either upon the arrival of a new packet, or at the onset of congestion where currently stored packets may then be discarded to accommodate a new, higher priority packet.

2. What information is used to make packet discard decisions? The main aspect is the granularity of the information, *i.e.*, is per flow buffer accounting done and used to discard packets from individual flows, or is only global, per class, information kept and used.

In order to be efficient, a buffer management mechanism should avoid violation of a service agreement by losing many high priority (conformant) packets during periods of congestion. Many buffer management mechanisms have been proposed and one of the most widely used is Random Early Discard (RED) [50], where, as a preventive measure, packets are discarded before the onset of congestion. RED relies on random dropping decisions when the buffer content exceeds a given threshold, so that heavy flows experience a larger number of dropped packets in case of congestion. Hence RED aims at penalizing flows in proportion to the amount of traffic they contribute, and therefore preventing any of them from grabbing a disproportionate amount of resources.

Complementing the use of non-FIFO queuing disciplines in an attempt to control the priority in which certain types of traffic is transmitted on a router interface, there are other methods, such as **admission control** and **traffic shaping**, which are used to control what traffic is actually transmitted into the network or the rate at which it is admitted. Leaky bucket and token bucket represent two important mechanisms for traffic shaping.

- *Leaky bucket.* Provides a mechanism by which bursty traffic can be shaped to present a steady stream of traffic to the network, as opposed to traffic with erratic bursts of low and high-volume flows. An analogy to understand how leaky bucket works is the case of a four-lane road where at a certain point all lanes converge into one; there, a regulated admission interval of traffic flow helps the traffic move. More formally, we can consider a leaky bucket as a reservoir of capacity c emptying at a rate r and filling due to the controlled input flow. Traffic conforms to the leaky bucket descriptor if the reservoir does not overflow and then satisfies the inequality $A(t) \leq rt + c$, where $A(t)$ is the amount of data generated by a flow in an interval of time t. The leaky bucket simplifies the problem of controlling input conformity, but its efficacy depends on being able to choose appropriate parameter rates for a given flow and then being able to efficiently guarantee QoS by means of admission control. Of course, if the volume of traffic is vastly greater than the bucket size, in conjunction with the drainage-time interval, traffic backs up in the bucket beyond capacity and is discarded. The leaky bucket may be viewed either as a statistical descriptor approximating the actual mean rate and burstiness of a given flow or as the definition of an envelope into which the traffic must be made to fit by shaping.

- *Token bucket.* The token bucket differs substantially from the leaky bucket. Whereas the leaky bucket fills with traffic and steadily transmits traffic at a continuous fixed rate when traffic is present, traffic does not actually transits the token bucket. The token bucket is a control mechanism that dictates when traffic can be transmitted

based on the presence of tokens (each of which represents a unit of bytes) in the bucket. The traffic description of a token bucket involves a bucket of "credits" or tokens, which provides opportunities for transmission. The bucket of size b (bytes) fills with tokens at a rate r (Bps), and data packets consume these tokens as they are transmitted. However, there is a peak rate p (Bps), and burst of data packets can be sent at this rate as long as enough tokens are available. At any time t, the source should not have sent more than $rt + b$. A meter can use this property to monitor the flow, and routers can police a flow by ensuring that packets do not violate this expression.

It should be noted that, when trying to provide QoS guarantees, it is necessary to combine the use of several of the mechanisms discussed above (*e.g.*, traffic shaping and scheduling). As Roberts notes [128], deterministic guarantees are possible if the amount of data $A(t)$ generated by a flow in an interval of length t satisfies a constraint of the form: $A(t) \leq \rho t + \sigma$. If the link serves this flow at a rate at least equal to ρ then the maximum buffer content from this flow is σ. Loss can therefore be completely avoided and delay bounded by providing a buffer of size ρ and implementing a scheduling discipline which ensures the service rate ρ [35]. The constraint on the input rate can be enforced by means of a mechanism such as a leaky bucket or a token bucket. In practice, it is usual to find implementations that shape traffic by means of a token bucket and then apply scheduling by using a variation of WFQ.

Furthermore, four principles for providing QoS guarantees in networks have been identified [81]:

1. Packet classification allows a router to distinguish among packets belonging to different classes of traffic.

2. It is desirable to provide a degree of isolation among traffic flows, so that one flow is not adversely affected by another misbehaving flow.

3. While providing isolation among flows, it is desirable to use resources (for example, link bandwidth and buffers) as efficiently as possible.

4. A call (connection) admission process is needed in which flows declare their QoS requirements and are then either admitted to the network (at the required QoS) or blocked from the network (if the required QoS cannot be provided by the network).

3.1.3 QoS in the IP world

Providing QoS in uni-application networks —such as the PSTN that carries only voice— is a difficult task that is alleviated by the fact that pertaining characteristics such as traffic load are well characterized and can be managed. For example, call arrival models have been made using the fact that calls arrive at a switch as a Poisson process, that is, the interarrival time between calls is drawn from an exponential distribution. Also, voice networks use circuit-switching, where admission control is made, a path is established, and

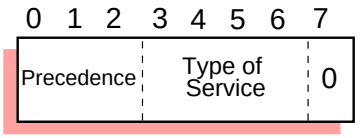

Figure 3.3: The IPv4 ToS header field.

sufficient reservations are made to service the call during the time the circuit is active. On the other hand, in multi-application or multi-service data networks (such as the Internet) the design concepts for traffic management used in voice networks can not be applied [114]. Traffic load in a data network is more variable; it has been shown [96; 34; 157] that traffic follows self-similar patterns, more like a fractal. Therefore, a multi-service network must be able to accommodate diverse requirements and must take a new approach for its service model to the customer.

The basic protocols of the Internet were not designed with QoS in mind. In fact, even basic traffic control mechanisms were initially absent. This situation has changed with time, as the protocols have grown richer with the incorporation of new functions. For instance, in the case of TCP [38], a congestion avoidance mechanism —pioneered by Van Jacobson [69]— was incorporated.

Since its early days, the Internet traditionally offered only one QoS (namely, best-effort) which assumes that all packets are equal and will be given the same treatment in the presence of network congestion. By 1981, when the definition of IP version 4 was published [37], early attempts to provide different levels of service were made —using a header field called type of service (ToS)— and three types of services were defined (see figure 3.3). This setup was later modified [5] to offer a fourth service; it was also stated that the 4 bits used for specifying the type of service were to be numerically additive rather than distinct entities. Because they were additive, the 4 bits provided a maximum of 16 possible values. In practice, however, implementations using the ToS field never really took off or were made incompatible with each other, thus making end-to-end QoS impossible.

Taking a next step in the evolution of IP QoS, the IETF is working on two different approaches for providing QoS on the Internet. One is the Integrated Services (IntServ) architecture [22; 132; 133], which requires applications to signal their service requirements to the network through a reservation request. IntServ currently uses the Resource Reservation Protocol (RSVP) as its end-to-end signaling protocol [23]. Unfortunately, the IntServ/RSVP architecture does not scale to the global Internet. The other approach, Differentiated Services (DiffServ) works in the core of the network through a scalable aggregated service mechanism [19; 105]. We will present detailed discussions for both approaches in the following sections.

3.2 The Integrated Services Approach

One of the first reactions of anyone analyzing the IETF's Integrated Services (IntServ) architecture [22; 156] is to be overwhelmed by its complexities. The architecture is in fact loosely inspired by mechanisms used in ATM —namely, in an effort to provision "guaranteed" services, as well as differing levels of best effort via a "controlled load" mechanism. It should be noted that, although IntServ provides signaling for QoS parameters (as ATM does), such signaling occurs at layer 3.

3.2.1 IntServ model and philosophy

One of the basic assumptions of the IntServ framework is that the basic underlying Internet architecture does not need to be modified in order to provide customized support for different applications but that, instead, a set of extensions can be developed that provide services beyond the traditional best-effort service. The IETF IntServ working group charter[2] articulates that efforts within the working group are focused on three primary goals [48]:

- *Clearly defining the services to be provided.* The first task faced by this working group was to define and document this "new and improved" enhanced Internet service model.
- *Defining the application service, router scheduling, and link-layer interfaces or "subnets".* The working group also must define at least three high-level interfaces: one that expresses the application's end-to-end requirements, one that defines what information is made available to individual routers within the network, and one that handles the additional expectations (if any) the enhanced service model has for specific link-layer technologies. The working group will define these abstract interfaces and coordinate with and advise other appropriate IP-over-subnet working groups efforts (such as IP-over-ATM).
- *Developing router validation requirements to ensure that the proper service is provided.* The Internet will continue to contain a heterogeneous set of routers, run different routing protocols, and use different forwarding algorithms. The working group must seek to define a minimal set of additional router requirements that ensure that the Internet can support the new service model. Instead of presenting specific scheduling and admission-control algorithms that must be supported, these requirements will likely take the form of behavioral tests that measure the capabilities of routers in the integrated services environment. This approach is used because no single algorithm seems likely to be appropriate in all circumstances at this time.

QoS in the context of the IntServ framework refers to the nature of the packet delivery service provided by the network, as characterized by parameters such as achieved bandwidth, packet delay, and packet loss rates [134]. Another basic assumption in the IntServ

[2] `http://www.ietf.org/html.charters/intserv-charter.html`

model is that resources in the network must be controlled in order to deliver QoS. Thus, all traffic must be subject to admission-control mechanisms, and provisions are made for a resource-reservation mechanism to grant resources guarantees. Point should be taken that, as stated in RFC 1633 [22], the term *guarantee* must be loosely interpreted in this context —guarantees must be *approximated* and *imprecise*. It is true that guarantee is not an approximate term, and that it implies an absolute state, so it should not be used to describe anything less; however, the IntServ model continues to define the guaranteed service level as a predictable service that a user can request from the network for the duration of a particular session.

The IntServ architecture consists of five key components: QoS requirements, resource-sharing requirements, allowances for packet dropping, provisions for usage feedback, and a resource reservation protocol (*i.e.*, RSVP).

3.2.2 QoS classes in IntServ

The IETF has considered various QoS classes such as Commited Rate [9], Protected Best Effort [61], Guaranteed [132], and Controlled Load [158], although to date only two of these, Guaranteed and Controlled Load service, have been formally specified for use with RSVP.

Guaranteed Service

Guaranteed Service provides an assured level of bandwidth, a firm end-to-end delay bound, and no queueing loss for conforming packets of a data flow. It is intended for applications with stringent real-time delivery requirements, such as certain audio and video applications that use "playback" buffers and are intolerant of any datagram arriving after their playback time. Each router characterizes the guaranteed service for a specific flow by allocating a bandwidth, R, and buffer space, B, that the flow may consume. This is done by approximating the "fluid model" of service [110; 111] so that the flow effectively sees a dedicated wire of bandwidth R between source and receiver. In a perfect fluid model, a flow conforming to a token bucket of rate r and depth b will have its delay bound by b/R provided $R \geq r$. To allow for deviations from this perfect fluid model in the router's approximation,[3] two error terms, C and D, are introduced; consequently, the delay bound now becomes $b/R + C/R + D$. However, with guaranteed service a limit is imposed on the peak rate, p, of the flow, which results in a reduction of the delay bound. In addition, the packetization effect of the flow needs to be taken into account by considering the maximum packet size, M. These additional factors result in a more precise bound on the end-to-end queueing delay as follows:

$$Q_{delayend2end} = \frac{(b-M)(p-R)}{R(p-r)} + \frac{M + C_{tot}}{R} + D_{tot} \qquad (\text{case } p > R \geq r) \qquad (3.2.1)$$

[3] Among other things, the router's approximation must take account of the medium-dependent behavior of the link layer of the data forwarding path

$$Q_{delayend2end} = \frac{M + C_{tot}}{R} + D_{tot} \qquad (\text{case } R \geq p \geq r) \tag{3.2.2}$$

where C_{tot} and D_{tot} represent the summation of the C and D error terms, respectively, for each router along the end-to-end data path.

In order for a router to invoke guaranteed service for a specific data flow, it needs to be informed of the traffic characteristics, `Tspec`, of the flow along with the reservation characteristics, `Rspec`. Furthermore, to enable the router to calculate sufficient local resources to guarantee a lossless service requires the terms C_{sum} and D_{sum}, which represent the summation of the C and D error terms, respectively, for each router along the path since the last reshaping point (see below).

`Tspec` parameters:
p = peak rate of flow (bytes/s)
b = bucket depth (bytes)
r = token bucket rate (bytes/s)
m = minimum policed unit (bytes)[4]
M = maximum datagram size (bytes)

`Rspec` parameters:
R = bandwidth, *i.e.*, service rate (bytes/s)
S = slack term (ms)

Guaranteed service traffic must be policed at the network access points to ensure conformance to the `Tspec`. The usual enforcement policy is to forward non-conforming packets as best-effort datagrams[5]; if and when a marking facility becomes available, these non-conforming datagrams should be marked to ensure that they are treated as best-effort datagrams at all subsequent routers.

In addition to policing of data flows at the edge of the network, guaranteed service also requires reshaping of traffic to the token bucket of the reserved `Tspec` at certain points on the distribution tree. Any packets failing the reshaping are treated as best-effort and marked accordingly if such a facility is available. Reshaping must be applied at any points where it is possible for a data flow to exceed the reserved `Tspec` even when all senders associated with the data flow conform to their individual `Tspec`s. Such an occurence is possible in the following two cases.

First, at branch points in the distribution tree where the reserved `Tspec`s of the outgoing branches are not the same, the reserved `Tspec` of the incoming branch is given by the "maximum"[6] of the reserved `Tspec`s on each of the outgoing branches. Consequently, some of the outgoing branches will have a reserved `Tspec` which is less than the reserved `Tspec` of the incoming branch; so it is possible that, in the absence of reshaping, traffic which conforms to the `Tspec` of the incoming branch might not conform when routed

[4]Policing will treat any IP datagram less that size m as being size m

[5]Action with regard to non-conforming datagrams should be configurable to allow for situations such as traffic sharing where the preferred action might be to discard non-conforming datagrams. This configuration requirement also applies to reshaping

[6]Maximum according to rules defined in [23]

through an outgoing branch with a smaller reserved `Tspec`. As a result, reshaping must be performed at each such outgoing branch to ensure that the traffic is within this smaller reserved `Tspec`.

Second, at merge points in the distribution tree for sources sharing the same reservation, the sum of the `Tspecs` relating to the incoming branches will be greater than the `Tspec` reserved on the outgoing branch. Consequently, when multiple incoming branches are each simultaneously active with traffic conforming to their respective `Tspecs`, it is possible that when this traffic is merged onto the outgoing branch it will violate the reserved `Tspec` of the outgoing branch. Hence, reshaping to the reserved `Tspec` of the outgoing branch is necessary.

Controlled Load Service

Unlike guaranteed service, controlled load service provides no firm quantitative guarantees. A `Tspec` for the flow desiring controlled-load service must be submitted to the router as for the case of guaranteed service, although it is not necessary to include the peak rate parameter. If the flow is accepted for controlled load service, the router makes the commitment to offer the flow a service equivalent to that seen by a best-effort flow on a lightly loaded network. The important difference is that the controlled load flow does not noticeably deteriorate as the network load increases. This will be true regardless of the level of load increase. By contrast, a best-effort flow would experience progressively worse service (higher delay and loss) as the network load increased. Controlled load service is intended for those classes of applications that can tolerate a certain amount of loss and delay provided it is kept to a reasonable level. Examples of applications in this category include adaptive real-time applications.

Routers implementing the controlled load service must check for conformance of controlled load data flows to thir appropriate reserved `Tspecs`. Any non-confoming controlled load data flows must not be allowed to affect the Qos offered to conforming controlled load data flows or to unfairly affect the handling of best-effort traffic. Within these constraints the router should attempt to forward as many of the packets of the non-conforming controlled load data flow as possible. This might be done by dividing the packets into conforming and non-conforming groups and forwarding the non-conforming group in a best-effort basis. Alternatively, the router may choose to degrade the QoS of all packets of a non-conforming controlled load data flow equally.

3.2.3 RSVP: a signaling protocol for IntServ

The IntServ model requires that nodes in the traffic path be able to support QoS control mechanisms; it also requires a mechanism by which the applications can communicate their QoS requirements to the nodes along the transit path, as well for the network nodes to communicate between one another the QoS requirements that must be provided for the particular traffic flows. This could be provided in different ways, but since another working group[7] within the IETF was developing a resource reservation setup protocol called RSVP [23; 159], this has traditionally been the chosen mechanism. It is worth

[7] http://www.ietf.org/html.charters/rsvp-charter.html

Message type	Description
`Path`	Sent from RSVP senders to receivers. They are used to store path information in each node in the traffic path.
`Resv`	Sent from receiver to sender along the same path followed by `Path` messages. They specify the desired QoS and set up the reservation state in each node in the traffic path.

Table 3.1: The two main types of RSVP messages

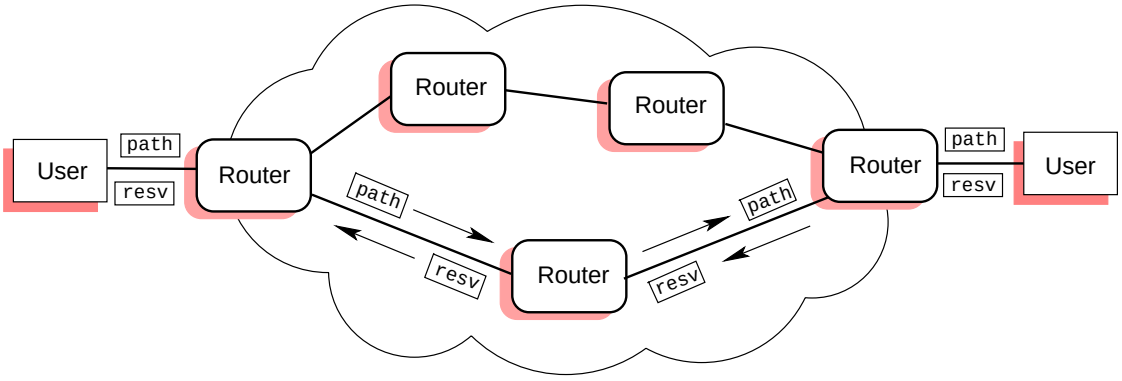

Figure 3.4: Flow of RSVP `Path` and `Resv` messages.

noting that some architecturally different [100] protocols such as ST-II [148] and ST-II+ [149] have been devised for similar purposes, but have not received much industry acceptance.

RSVP is a signaling protocol that provides reservation setup and control, which is intended to provide the closest thing to circuit emulation on IP networks. It represents the biggest departure from standard best-effort IP service and provides the highest level of QoS in terms of service guarantees, granularity of resource allocation and detail of feedback to QoS-enabled applications. As such, it is the most complex of the QoS technologies for the IP environment. RSVP identifies a communication session by the combination of destination address, transport-layer protocol type, and destination port number. It is important to note that each RSVP operation only applies to packets of a particular session; therefore, every RSVP message must include details of the session to which it applies.

Reservations in RSVP are *receiver-initiated* and two main types of messages (with several sub-types) are used: `Path` and `Resv`. A brief description of these messages is given in table 3.1. RSVP request only unidirectional resources —resource reservation requests are made in one direction only. Although an application can act as a sender and receiver at the same time, RSVP treats sending and receiving as logically distinct functions. RSVP establishes and maintains a *soft state* by sending periodic refresh messages along the data path to maintain the reservation and path state.

RSVP is not a routing protocol, it is merely used to reserve resources along the existing route set up by whichever underlying routing protocol is in place. RSVP messages can be transported "raw" within IP datagrams using protocol number 46, although hosts without this raw input/output capability may first encapsulate RSVP messages within a UDP header. The following is a simplified overview of how RSVP works (depicted in figure 3.4):

- Senders characterize outgoing traffic in terms of the upper and lower bounds of bandwidth, delay, and jitter. RSVP sends a `Path` message from the sender that contains this traffic specification (`Tspec`) information to the (unicast or multicast receiver(s)) destination address. Each RSVP-enabled router along the downstream route establishes a "path-state" that includes the previous source address of the `Path` message (*i.e.*, the next hop "upstream" towards the sender).

- To make a resource reservation, receivers send a reservation request (`Resv`) message "upstream". In addition to the `Tspec`, the `Resv` message includes a request specification (`Rspec`) that indicates the type of Integrated Services required —either Controlled Load or Guaranteed— and a filter specification (`Filterspec`) that characterizes the packets for which the reservation is being made (*i.e.*, the transport protocol and port number). Together, the `Rspec` and `Filterspec` represent a flow descriptor that routers use to identify each reservation (also known as a "flow" or a "session").

- When each RSVP router along the upstream path receives the `Resv` message, it uses the admission control process to authenticate the request and allocate the necessary resources. If the request can not be satisfied (due to lack of resources or authorization failure), the router returns an error back to the receiver `ResvErr`. If accepted the router sends the `Resv` upstream to the next router.

- When the last router[8] receives the `Resv` and accepts the request, it sends a confirmation message back to the receiver.

- There is an explicit tear-down process for a reservation when sender or receiver ends a RSVP session (`ResvTear` and `PathTear`).

We have already mentioned that RSVP is pretty complex, so in the above description we have obviously omitted a lot of details such as:

- `Adspec`, which is an optional object that the sender may include in `Path` messages in order to advertise to receivers the characteristics of the end-to-end communications path.

- Reservation styles, which deal with how one reservation interacts with others.

- `Filterspec`, which allows characterization of "sub-flows" that could be used in a hierarchically encoded signal for heterogeneous receivers, for example.

[8] the "last router" is either the closest to the sender or at a reservation merge point for multicast flows

- Policy data, which provides detailed condition information for use in resource reservation policy decisions.

3.2.4 Problems with RSVP/IntServ

As already mentioned, RSVP provides the highest level of IP QoS available. It also allows an application to request QoS with a high level of granularity and with the best guarantees of service delivery possible. That makes one wonder why some other approach should be needed for IP QoS. The reason lies in the many problems that RSVP/IntServ have shown, for instance:

- Its complexity and overhead in routers.
- Per-flow state is O(n), so it is not scalable when the numbers of flows grow large, as may be the case in backbones.
- It needs policy controls and support for accounting and security. The RSVP Admission Policy (RAP) working group is already addressing some of these issues.
- Sometimes control/notification traffic is needed from the sender. But since it is receiver-based, question such as: which receiver pays for the shared part of the tree? arise.
- It is soft-state based, so route/path pinning (stability) is needed. Thus the number of changes during a session should be minimized.
- Throughput and delay guarantees require support of lower layers. In some cases, such as when using shared Ethernet (or even worse, wireless), it will not be possible to enforce Guaranteed or Controlled Load services. Thus, more suitable lower layer technologies such as switched full-duplex LANs are necessary.

It is then clear that RSVP/IntServ may be overkill for many applications and for some portions of the network. Simpler, less fine-tuned methods are needed, and that is what DiffServ provides, as will be shown in the next section.

3.3 The Differentiated Services Approach

In early 1998, the IETF formed the differentiated services (DiffServ) working group[9] to take another approach for the problem of IP QoS, and to try to overcome the shortcomings found with IntServ. DiffServ minimizes signaling and concentrates on aggregated flows and per-hop behavior (PHB) applied to a network-wide set of traffic classes. To build differentiated services, two main items are needed: servers and routers that apply appropriate traffic classification and metering; and mechanisms for defining, managing, and delivering service policy to the routers and servers [27].

[9] `http://www.ietf.org/html.charters/diffserv-charter.html`

3.3.1 Design philosophy

Several factors have driven the design of DiffServ, among them we can mention ([32]):

- The solution has to scale. To achieve this, individual host-to-host microflows are aggregated into a single larger aggregate flow which then will receive special treatment.
- The solution should be applicable to all applications and should not require a special control protocol or new application programming interfaces as is the case with RSVP.
- Even though router and switch technologies are advancing rapidly,[10] they do not need to be burdened with the instantiation of per-flow or per-customer state. A more efficient and scalable option is to provision per-class or per-service state.
- ISPs (Internet Service Providers) are desperate to offer a portfolio of services their customers will pay for, being QoS one of them.

As stated in the DiffServ working group objectives, "there is a clear need for relatively simple and coarse methods of providing differentiated classes of service for Internet traffic, to support various types of applications, and specific business requirements. The differentiated service approach to providing quality of service in networks employs a small, well-defined set of building blocks from which a variety of aggregate behaviors may be built. A small bit-pattern in each packet, in the IPv4 ToS octet or the IPv6 Traffic Class octet, is used to mark a packet to receive a particular forwarding treatment, or per-hop behavior, at each network node. A common understanding about the use and interpretation of this bit-pattern is required for inter-domain use, multi-vendor interoperability, and consistent reasoning about expected aggregate behaviors in a network. Thus the working group has standardized a common layout for a six-bit field of both octets called the DS field. RFC 2474 [105] and RFC 2475 [19] define the architecture, and the general use of the bits within the DS field (superseding the IPv4 ToS octet definitions of RFC 1349 [5])". Figure 3.3 shows the original IPv4 ToS field layout, and figure 3.5 shows the DS field with the layout of the Differentiated Services Code Point (DSCP).

3.3.2 Architectural elements

One of the main tenets of DiffServ is its distinction between the edge and the core (middle) of an administrative domain. IntServ performs classification and policing on all packets matching a reservation in every router along the path. In contrast, DiffServ pushes most of the classification and policing functions to the edges of the administrative domain and simplifies the forwarding functions (PHBs in DiffServ parlance) in the core of the domain[11]. When an edge router receives a packet from a host or neighboring domain,

[10] OC-48 (2.4 Gbps) line rates are currently supported and OC-192 (10 Gbps) rates are coming soon

[11] Note how this complies with end-to-end arguments[129] that are the core of the Internet philosophy

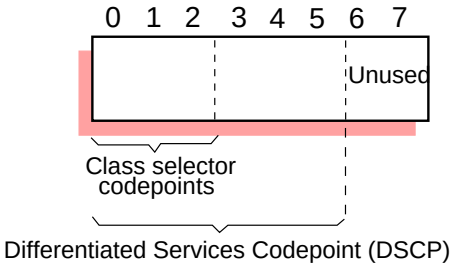

Figure 3.5: The DS octet with its DSCP sub-field.

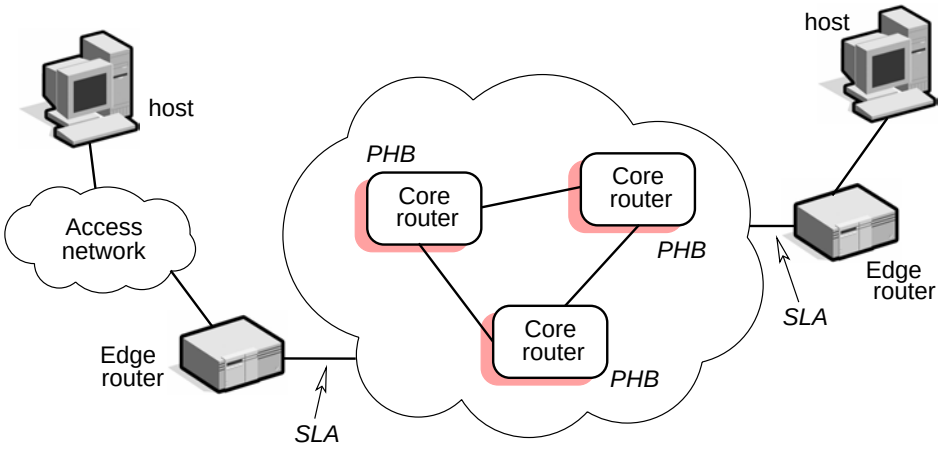

Figure 3.6: Elements of a DiffServ architecture.

it will usually fully classify and police each packet. In the core network, however, no policing occurs and classification is reduced to examining the DS field[12] in the IP header to determine the appropriate forwarding action. Diffserv uses the notion of a DS domain for its traffic-conditioning agreement (TCA) operations. It is a network or a collection of networks operating under an administration with a common provisioning policy. It is responsible for meeting the service-level agreement (SLA) between the user and the DS domain service provider.

For instance, in order for a customer to receive differentiated services from its ISP, it must have a SLA which basically specifies the service classes supported and the amount of traffic allowed in each class [160]. A SLA can be static or dynamic. *Static SLA*s are negotiated on a regular (*e.g.*, monthly, yearly) basis. Customers with *dynamic SLA*s must use a signaling protocol (*e.g.*, RSVP) to request for services on demand. Customers can mark DS fields of individual packets to indicate the desired service or have them marked by the leaf router based on a multi-field classification. At the ingress of the ISP networks, packets are classified, policed, and possibly shaped. The classification, policing and shaping rules used at the ingress routers are derived from the SLAs, as is the amount of buffering space needed from these operations. When a packet enters one domain from

[12]The process of sorting packets based on the contents of the DS field is called Behavior Aggregate (BA) classification

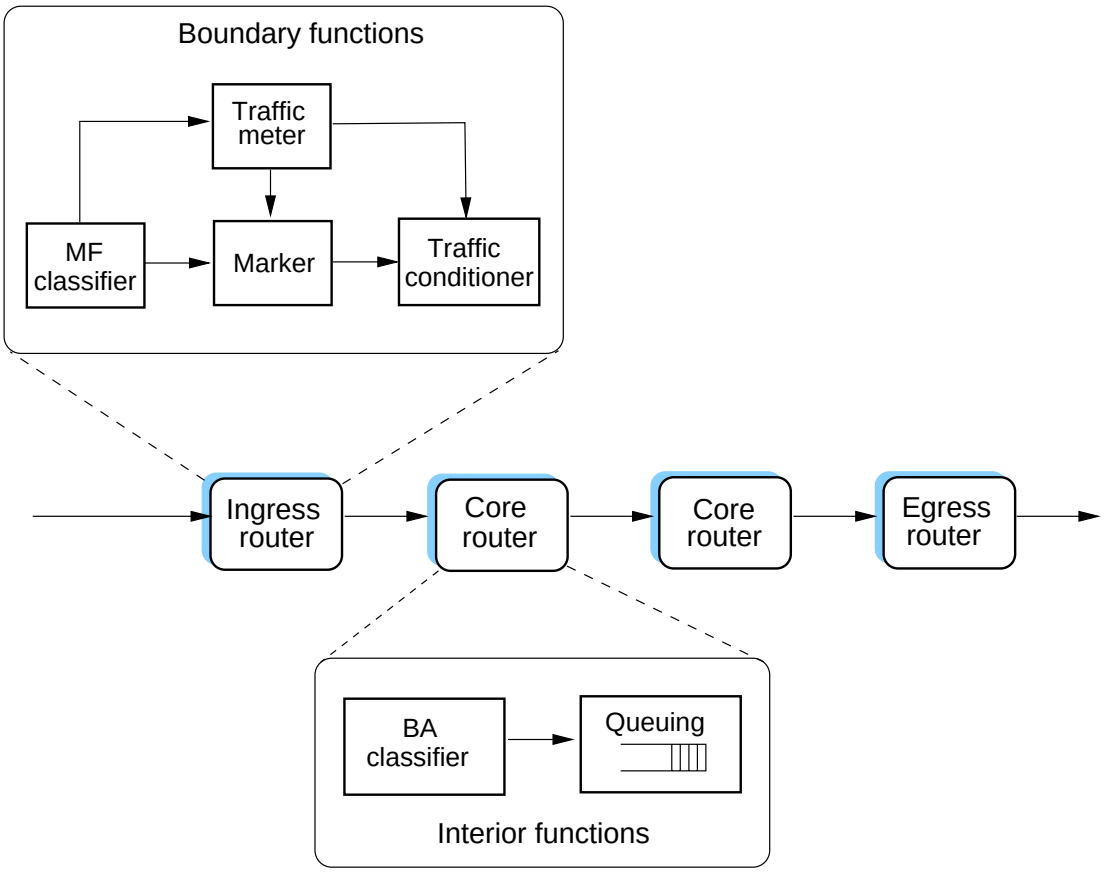

Figure 3.7: QoS mechanisms in a DiffServ architecture.

another domain, its DS field may be re-marked as determined by the SLA between the two domains. Using the classification, policing, shaping and scheduling mechanisms, any services can be provided, for example 1) *Premium Service* for applications requiring low delay and low jitter service; 2) *Assured Service* for applications requiring better reliability than best-effort; and 3) *Olympic Service*, which provides three tiers of services: Gold, Silver, and Bronze, with decreasing quality [106; 62]. Note that the Differentiated Services only define DS fields and PHBs; it is the ISP's responsibility to decide what services to provide. Further details [155] on the mechanisms for providing these services follow, and figure 3.7 show how they fit within a DiffServ architecture.

- **Classifying**. A router must be able to look at each packet and identify the flow to which it belongs. Examining the source and destination IP addresses and, possibly, the source and destination port numbers —for UDP and TCP— provides a means for uniquely identifying the flow. This process of flow identification is commonly referred to as classification.

- **Metering**. After a flow is classified, to determine that the flow is not exceeding the agreed resource consumption limits, a router must first measure a flow's volume over some period of time. However, measuring a flow rate is not sufficient to determine a

flow's compliance. Suppose that the host may choose to send 5 Mb per minute. The host may choose to send 5 Mb over a period of 5 seconds at 30-second intervals rather than sending 166 Kbps continuously. Sending higher rates of traffic at intervals is also referred to as sending bursts; when a host sends bursts the resulting flow is said to be bursty. A typical QoS agreement often defines limits on the size of busts as well as the maximum bandwidth for a flow. Consequently a router must measure the flow rate as well as the size of traffic bursts. Measuring traffic is also important for billing the end user or the company for the various services they use. When different services are offered over the same physical network connection, the need to account and bill for these services individually becomes important. In some cases the destination may also factor into the charges for the service if the distances involved are great. Taken together, the measurement of flow rate and burst size is often referred to as metering.

- **Shaping**. When a flow contains a burst of packets, a router can choose to process the burst in a number of ways. One alternative is to process it normally if it falls within some predefined or negotiated limit. Another alternative is to absorb the burst and pace the packets out over a longer period of time. This pacing of the burst smooths or even eliminates it altogether. The last alternative is to drop the packets in the burst that exceed a particular threshold. This threshold could be an upper limit on the number of packets in the burst that the router will have to retain at any point in time. Another threshold could be the amount of time that the packets would have to be kept before the data becomes stale. The process of holding bursts and pacing the traffic is called shaping.

- **Dropping**. When a flow exceeds the negotiated rate or a burst exceeds a maximum threshold, a router may choose to drop one or more packets in the flow. Dropping packets is a common practice for controlling congestion. Depending on the service, the criteria for dropping packets vary. When QoS is taken into consideration, determining whether a packet should be dropped can be a relatively complicated decision.

- **Policing**. It is worth noting that the metering, shaping, and dropping are collectively referred to as policing.

3.3.3 Per-hop behaviors

The IETF has recognized that PHBs must be defined for DiffServ nodes to support a diverse user community; to that end, work is being done on two PHBs: expedited forwarding (EF) [70] and assured forwarding (AF) [62].

- **EF**. This PHB was designed to support low loss, low delay, and low jitter connections. It appears as a point-to-point virtual leased line (VLL) service between endpoints with a peak bandwidth. To minimize jitter and delay, packets must spend little or no time in router queues. Therefore the EF PHB requires that the traffic be conditioned to conform to the peak rate at the boundary, and the network of routers

be provisioned such that this peak rate is less than the minimum packet departure rate at each router in the network. The EF PHB uses a single DSCP bit to indicate that the packet should be placed in a high-priority queue on the outbound link of each router hop. The code-point for EF is 101110. The DS traffic conditioning block must treat the EF PHB as the highest priority of all traffic. However, EF packets are not allowed to preempt other traffic. Consequently a tool, such as a token bucket, must be part of the DS features. RFC 2598 [70] includes an appendix (Appendix A) that explains the results of some simulations of models to support the EF PHB.

- **AF**. This PHB defines four relative classes of service with each service supporting three levels of drop precedence. Twelve distinct DSCP bit combinations define the AF classes and the drop precedence will be discarded ahead of those with a lower drop precedence. The four AF classes define no specific bandwidth or delay constraints other than that AF class 1 is distinct from AF class 2, and so on.

3.4 Chapter Conclusions

Granting QoS guarantees involves several elements such as admission control, traffic shaping and scheduling, among others. We have seen that service guarantees are possible if a flow is constrained by applying a traffic shaping mechanism (such as a token bucket) and implementing a scheduling discipline to service packets (for instance, a variation of WFQ).

The Internet protocols were not designed with QoS in mind from the beginning. However, recent efforts propose the provision of QoS based on different models, where the most prominent ones are the Integrated Services (IntServ) and the Differentiated Services (DiffServ) models. IntServ is a somewhat complex model that aims for the provision of guaranteed services through signaling and reservations. It has proved to have some serious problems in terms of scalability, complexity, and overhead. The DiffServ model provides a more flexible approach where only statistical guarantees are provided. It minimizes signaling and concentrates on aggregated per-flow behaviors applied to a network-wide set of traffic classes. Differentiated services are based on routers applying traffic classification and metering, as well as on mechanisms for defining, managing, and delivering service policy to the routers.

Mobility Management in IP Networks

Chapter 4

Mobility Management in IP Networks

Push on, keep moving

— Thomas Morton - A Cure for the Heartache. Act ii. Sc. 1.

Contrary to systems like GSM [124], where mobility has been an integral part of the system since the very beginning, the Internet was born at a time when the notion of mobile networking devices was non-existent. Therefore, there has been a complete lack of support for mobile devices on the Internet for a long time. However, proposals for providing mobility support at the IP layer have been made, Mobile IP being the most prominent of these.

Mobile IP allows mobility of devices, potentially around the world; this is why the type of mobility support it provides is sometimes referred to as global mobility. However, as we will see, mobility within a limited geographical area (called micro-mobility), has different characteristics and requirements that pose the need for specialized support. In this chapter we will discuss global mobility, with a particular emphasis on Mobile IP. We will also discuss how other recent proposals such as Hawaii and Cellular IP address the particular problems of micro-mobility. Seamless mobility and context transfers for heterogeneous networks is a related topic, and we will overview the work in progress within the IETF's SeaMoby working group on that topic. A final section presents recent proposals to provide QoS in mobile environments. We will see how the Insignia project incorporates in-band signaling for QoS reservations. A signaling protocol for reservations, this time based on the IntServ model, is also explored with CLEP and MIR. Then, in order to compare with an approach that is not based on IP, we will see how Wireless ATM tries to extend ATM's QoS capabilities to mobile networks.

It is important to note that the type of mobility we are interested in is, for example, the case when a user is connected using several applications across the Internet, changing eventually its point of attachment in a dynamic way, and maintaining all open connections despite the change. This is in contrast to a user with a portable computer who travels abroad and, when arriving at its destination, turns on his computer to connect to an ISP, obtaining a new address in the process (possibly via DHCP [44; 4]). This latter case is better known as "nomadic" networking.

4.1 Global Mobility and Mobile IP

4.1.1 IP-based mobility challenges

Even though networking-enabled mobile devices are becoming more common everyday, most networking protocols —including the TCP/IP protocol suite— have been designed under the tacit assumption that hosts are always attached to the network at a single physical location. Therefore, host mobility is seemed as a rarely occurring fact that can be handled manually. Consider for instance the following scenario: a university researcher is usually connected to the network at his office's desk, but occasionally needs to take his laptop computer with him to teach some courses; the classrooms may be somewhere else in the same building or even in a different building. If the researcher's desk and the classroom have direct access to the same IP subnet, then the mobility process is trivial. In situations where this is not the case, the only solution is for the user to acquire a new IP address from the appropriate local authority. Then, several configuration files on the moving machine, on various name servers and on other machines that use the original IP address to identify the moving machine, need to be modified. Thus, moving the computer from one place to another involves a slow, error prone, manual procedure that a typical user does not have the skills or is not willing to carry out. Moreover, even if the process is successfully performed, the mobile host will lose its former identity and will usually need re-booting.

The situation is then that, given TCP/IP's early design assumptions that end systems are stationary, if during an active connection one end system moves, then the whole connection breaks, obviously disrupting all networking services layered on top of TCP/IP. There are two approaches for solving the problem:

1. To completely redesign (or start a design from scratch of) internetworking protocols with the specific goal of supporting mobile end systems, or
2. to provide additional services at the network layer in a backward compatible manner which make mobile internetworking possible.

Although the first approach is undoubtedly interesting from a research point of view, its practical application is not feasible, since it requires radical changes to the currently deployed networking infrastructure. It consists on changing TCP/IP, which forms the fabric of today's world-wide Internet. Therefore, the proposals we will present ahead are based on the second approach.

Evidence has been given [13] that in order to retain transport layer connections, a mobile host's address must be preserved regardless of its point of attachment to the network. The problem with a transport layer protocol such as TCP is that, for instance, a TCP connection is identified by a 4-tuple:

```
<source IP address, source TCP port, destination IP address, destination port>
```

So, if neither host moves, all elements of the tuple remain fixed and the TCP connection can be preserved. However, if either end of the connection moves, the following problem will take place:

- If the mobile host acquires a new IP address, then its associated TCP connection identifier also changes. This causes all TCP connections involving the mobile host to break.

- If the mobile retains its address, then the routing system cannot forward packets to its new location.

These problems come from the very design of IP, which, in addition to fragmentation and re-assembly, is responsible for "providing the functions necessary to deliver a package of bits (an Internet datagram) from a source to a destination over an interconnected system of networks" [37]. So, this definition designates responsibility to IP for routing datagrams to and from mobile hosts transparently to higher layers. The problem is that IP addresses serve a dual purpose, as they are not only used by higher layers to identify source and destination hosts, but also, by their division into network and host parts, also contain location information. Therefore, in its role as an identifier, *an IP address must be constant during mobility to avoid affecting higher layers.*

Research studies on IP mobility have suggested that mobility is essentially an address translation problem and is best resolved at the network layer [13]. As Figure 4.1 shows, a mobile host MH can move away from its home network and attach to the Internet through a foreign network. While away, MH obtains a forwarding address derived from the address space of the foreign network. However, if another host S tries to send packets to MH, it will do so using MH's home address. The problem is resolved by the use of an Address Translation Agent (ATA) at the home network, and a forwarding agent (FA) at the foreign network. These agents perform functions f and g, respectively, that are defined as follows:

- $f : homeaddress \longrightarrow forwardingaddress$

- $g : forwardingaddress \longrightarrow homeaddress$

This way, when S sends packets to MH, they first pass through ATA. This agent performs mapping f to send the packets to the address that MH acquired in the foreign network. At the foreign network, FA intercepts all packets containing MH's forwarding address. It then proceeds to apply the function g to map from this forwarding address to MH's original home address and effectively forward the packets.

Some other requirements for a IP-based mobile host protocol have been identified [102]: *Operational transparency* is an essential requirement, meaning that a user does not have to perform any special actions, such as manual reconfiguration, before, during, or after host migration (mobility). This can only be achieved by providing mechanisms for *migration detection* and to perform the appropriate actions to ensure continuing network services from all hosts to the mobile host's new location. An additional requirement is that of *performance transparency*, meaning that the use of a mobile host protocol should not affect the performance of applications in mobile hosts, which should be comparable to that found when the same applications run on fixed hosts. Factors that ensure performance transparency include optimum routing of packets to and from mobile hosts, efficient and

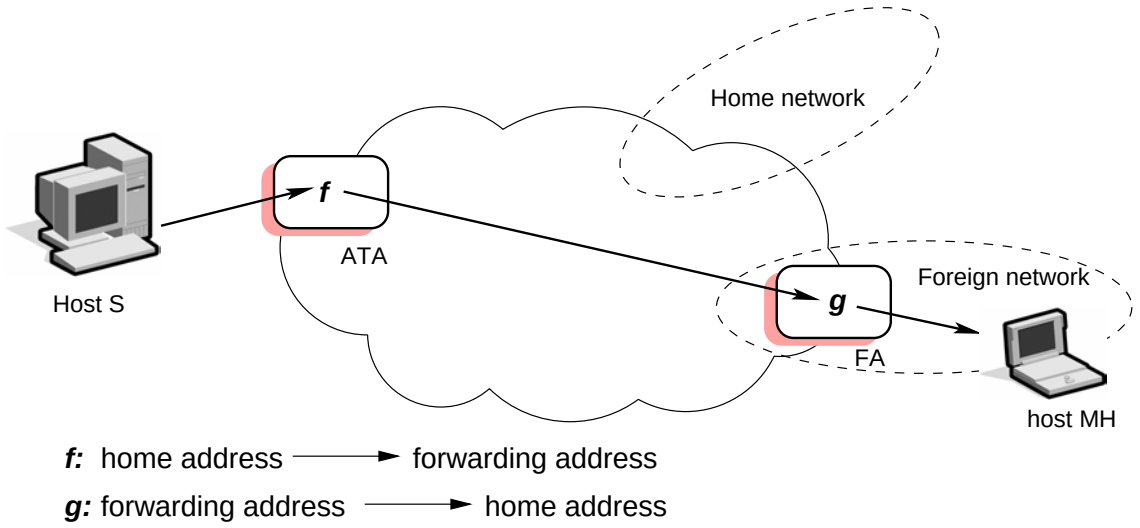

Figure 4.1: Mobility as an address translation problem.

robust migration procedures and efficient use of network resources such as transmission and processing bandwidth. Other factors should be considered, being an important one that of *backward compatibility*, as mentioned before.

4.1.2 Prior work

Before discussing how Mobile IP handles mobility management, we will give a brief overview of prior proposals that in some way or another influenced the design of Mobile IP. There are several papers detailing these protocols [136; 121; 102; 144; 67; 65], and here we will just give brief descriptions of them.

Mobile*IP

This protocol uses a virtual mobile subnet created by placing a small number of cooperating mobile subnet routers (MSRs) wherever mobile nodes may be connected to the network. When a mobile node moves around to a new location, it registers with a nearby MSR, and then informs the previous MSR about its current location. A packet for the mobile node is delivered by the MSR. If this MSR is not the local one to this mobile node, the datagram is encapsulated (using the IPIP method), and forwarded to the local MSR that will then decapsulate it and send it to the mobile node.

The above scheme is good for narrow range mobility. But when it comes to wide range mobility, it has limitations. To overcome this problem, a "popup" operation was defined for a wide area mode. The idea is as follows: when a mobile node moves, it registers with a MSR on its home network, and also acquires a temporary address. So when the MSR receives datagrams addressed for the mobile node, it encapsulates datagrams and tunnels them directly to the mobile node's temporary address.

VIP

In this protocol, when a mobile node moves to a new location, it acquires a temporary address from its local address server. Then, a Propagating Cache method is used to distribute the mappings between the temporary address and the identifier information, through the network. The mapping is also sent to the mobile node's home gateway. As a datagram is passed through the network to the mobile node, the intermediate gateways use the source and destination mappings to update their caches. Various methods are defined, such as cache time-outs, and management procedures, to prevent stale information from being held for extended periods. If an intermediate node gets a packet and knows the mobile node's current location it directly forwards it to the mobile node; otherwise it will send the packet to the mobile node's home gateway, which always knows the location of the mobile node, and it forwards the packet to the destination.

IBM I

This protocol uses loose source routing (LSR) to propagate the mapping between location information and identifier information through the network, so an optimal route can be taken when the packet is sent. The basic operation of the protocol is as follows: when a mobile node moves to a new location, it detects the change in position, registers with a base station, which is similar to Mobile*IP's MSR, and informs its home mobile router (MR), which is similar to VIP's home gateway. When the mobile host migrates, it notifies its previous base station and the MR of the new location. When a correspondent node sends a datagram to the mobile node through the old base station, it will be forwarded to the MR for the correct routing. LSR in the correspondent node will be eventually updated to make sure route optimization is performed.

IBM II

The architecture suggested in IBM II is similar to IBM I. The main difference is that IBM II allows the use of encapsulation. This implies that every packet sent from the correspondent node must be routed via the MR (which is now called RDS/LD). Therefore, unlike IBM I, routing optimization is not used.

MIP

When a mobile node moves to a new location, it registers with an Internet Access Point (IAP) that acts as its local agent. Then, it notifies its location directory (LD), responsible for making bindings for its mobile nodes available to a home redirector (HR) that serves the mobile node's home network. The packet can always be forwarded to the HR which always has a binding for the mobile node when the mobile node's current location is not known by any of IAPs or correspondent node that receive the data packet. There is another entity in this protocol called a Mobile Support Router (MSR) which is similar to an IAP except that it only keeps non-local binding. It is used mainly to optimize communications between mobile nodes.

MHRP

This protocol uses ideas similar to loose source routing. However, it uses a new encapsulation method compatible with ICMP with a header of 8 or 12 bytes to reduce overhead.

Mobile protocol proposals, like the ones we have presented, have been suggested by

various researchers over several years. Some of these proposals have found their way into the Mobile IP specification. Mobile*IP and VIP came out around the same time. For Mobile*IP, its most known features are virtual mobile subnet, and packet encapsulation; for VIP, are mobile host locations in special routers, and tunneling using a new IP option. IBM MIP uses loose source routing to deal with the mobile node trace issue existing in mobility problems. MHRP adopts a new type of encapsulation different than an old IP option which is defined in VIP. We will now discuss Mobile IP, which is the culmination of several of these efforts.

4.1.3 Operation of Mobile IP

Convinced that mobile networking was not just a passing fad, but an unstoppable trend that would increasingly overtake IP networking, the IETF created the Mobile IP working group[1]. The basic Mobile IP standard [118] specifies a mobility management architecture for the Internet. In principle, both local-area and wide-area mobility across wired and wireless networks can be handled, although certain inefficiencies have been detected. We will later see extensions to Mobile IP proposed to overcome such inefficiencies. In order to properly understand the operation of Mobile IP, we will define several entities and terms [130] as defined in RFC 2002 [118].

- *Mobile node (MN).* A mobile node is an end-system or a router that can change its point of attachment to the Internet using Mobile IP. The MN keeps its IP address and can continuously communicate with any other system in the Internet as long as link-layer connectivity is given. Moreover, nodes are not necessarily small devices such as laptops with antennas or mobile phones; a router onboard an aircraft can be a powerful mobile node.

- *Correspondent node (CN).* At least one partner is needed for communication. In the following the CN represents this partner for the MN. The CN can be a fixed of mobile node.

- *Home network.* The home network is the subnet the MN belongs to with respect to its IP address. Within the home network no Mobile IP support is needed.

- *Foreign network.* The foreign network is the current subnet the MN visits, that is not the home network.

- *Foreign agent (FA).* The FA can provide several services to the MN during its visit in the foreign network. The FA can have the COA (defined below) thus acting as tunnel endpoint and forwarding packets to the MN. Furthermore, the FA can be the default router for the MN. An FA can also provide security services as it belongs to the foreign network, as opposed to the MN only visiting. For Mobile IP functioning, an FA is not strictly needed. Typically, an FA is implemented on a router for the subnet the MN attaches to.

[1] http://www.ietf.org/html.charters/mobileip-charter.html

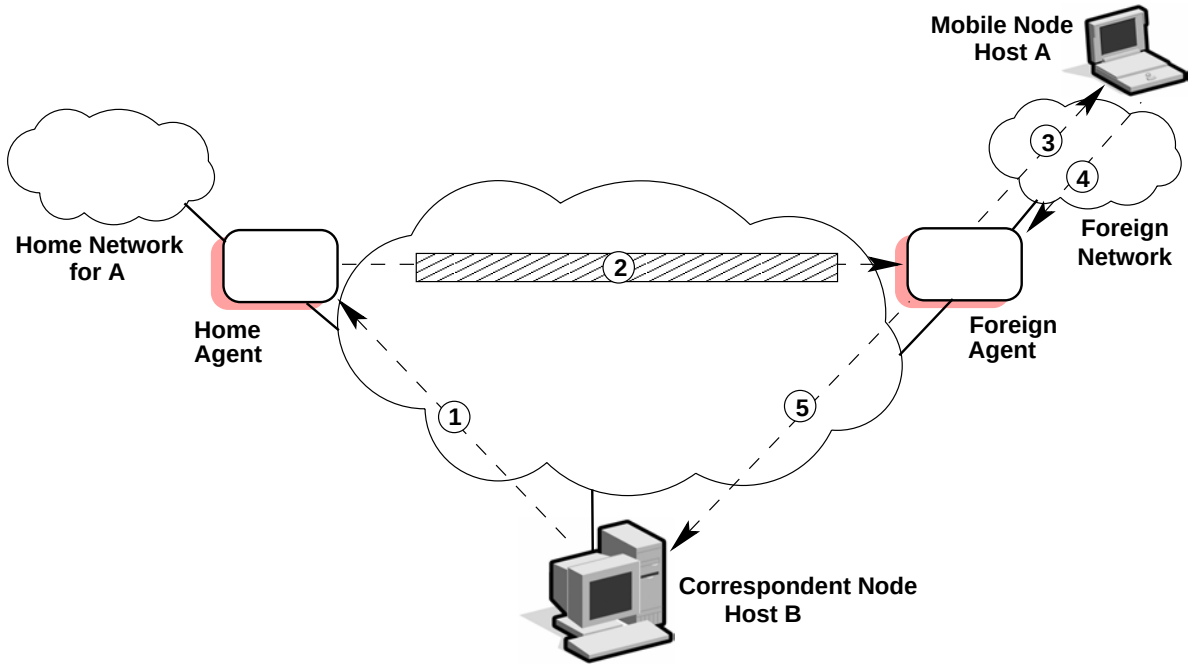

Figure 4.2: Basic Mobile IP scenario.

- *Care-of address (COA).* The COA defines the current location of the MN from an IP point of view. All IP packets sent to the MN are delivered to the COA, not directly to the IP address of the MN. Packet delivery toward the MN is done using a tunnel endpoint, *i.e.*, the address where packets exit the tunnel. There are two different possibilities for the location of the COA:
  - *Foreign agent COA.* The COA could be located at the FA, *i.e.*, the COA is the IP address of the FA. Thus the FA is the tunnel end-point and forwards packets to the MN. Many MN using the FA can share this common COA.
  - *Co-located COA.* The COA is called co-located if the MN temporarily acquires an additional IP address that acts as COA. This address is now topologically correct and the tunnel endpoint is at the MN. Co-located addresses can be acquired using services such as DHCP. One problem associated with this approach is the need for many additional addresses if many MNs request a COA. This is not always a good idea considering the scarcity of IPv4 addresses.

- *Home agent (HA).* The HA, located in the home network, provides several services for the MN. The tunnel for packets toward the MN starts at the HA. Furthermore, the HA maintains a location registry, *i.e.*, it is informed of the MN's location by the current COA. Three alternatives for the implementation of an HA exist.

- The HA can be implemented on a router that is responsible for the home network. This is obviously the best position, because without optimizations to Mobile IP, all packets for the MN have to go through the router anyway as explained later.

- If changing the router's software is not possible, the HA can be implemented on an arbitrary node in the subnet. A disadvantage of this solution is the double crossing of the router by the packet if the MN is in a foreign network. A packet for the MN comes in via the router; the HA sends it through the tunnel which again crosses the router.

- Finally, all MNs are always in a foreign network. The HA could be on the 'router' only acting as a manager for MNs belonging to a virtual home network.

Figure 4.2 shows the basic operation of Mobile IP. A mobile node is normally attached to its home network using a static home address. When the mobile node moves to a foreign network, it makes its presence known by registering with a foreign agent. The mobile node then communicates with a home agent in its home network, giving it the care-of address, which identifies the foreign agent's location. Typically, routers in a network will implement the roles of home and foreign agents. When IP datagrams are exchanged over a connection between the mobile node A and a correspondent host B, the following operations occur [139]:

1. Host B transmits an IP datagram destined for mobile node A, with A's home address in the IP header. The IP datagram is routed to A's home network.

2. At the home network, the incoming IP datagram is intercepted by the home agent. The home agent encapsulates the entire datagram inside a new IP datagram, which has A's care-of address in the header, and retransmits the datagram. The use of an outer IP datagram with a different destination IP address is known as tunneling.

3. The foreign agent strips off the outer IP header, encapsulates the original IP datagram in a MAC-level PDU (for example, an Ethernet frame), and delivers the original datagram to A across the foreign network.

4. When A sends IP traffic to B, it uses B's IP address. In our example, this is a fixed address; that is, B is not a mobile node. Each IP datagram is sent by A to a router on the foreign network for routing to B.

5. The IP datagram from A to X travels directly across the Internet to B, using B's IP address.

Three basic capabilities are needed to support the operations above: discovery, registration, and tunneling. We will now give details on them.

Agent advertisement and discovery

When a MN moves, several questions arise: how does the MN discovers it has moved? and then, how does it find a foreign agent? These types of problems are solved by having foreign and home agents to periodically advertise their presence using messages that can be seen as a beacon broadcast into the subnet. The *discovery* process has been built on top of an existing standard protocol, Router Advertisement, specified in RFC 1256 [39], which is not modified but just extended to integrate mobility functions. Routers in the

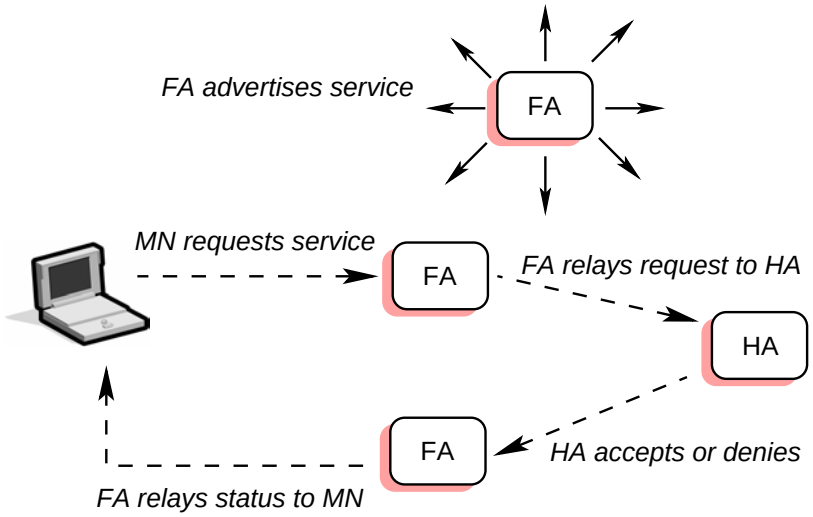

Figure 4.3: Registration operations in Mobile IP.

fixed network implementing this standard also advertise their routing service periodically to the attached links. When the router advertisements are extended to also contain the needed care-of address (COA), they are known as *agent advertisements*.

A mobile node listens for agent advertisement messages that are emitted periodically. Because a FA can be on the home network of the MN (set up to serve visiting MNs), the arrival of an agent advertisement does not necessarily tell the MN that it is on a foreign network. The MN must compare the network prefix of the router IP address with the network prefix of its own home address. If these network prefixes do not match, then the MN is on a foreign network. FAs (and HAs) are expected to periodically issue agent advertisement messages. However, if a MN needs a COA and does not wish to wait for the periodic advertisement, the MN can broadcast or multicast a solicitation that will be answered by any FA of HA that receives it. Thus, an agent advertisement has the following functions:

- allows for the detection of mobility agents,
- lists one or more available COAs,
- informs the MN about special features provided by FAs, for example, alternative encapsulation techniques,
- lets MNs determine the network prefix and status of their link to the Internet, and
- lets the MN know whether the agent is a HA, a FA, or both, and therefore whether it is on its home network or a foreign network.

Using a standard such as RFC 1256 for something different than the original purpose of router advertisements causes some problems. A quite obvious one is the minimum

interval of 3 seconds between advertisements, which for a wired network where the topology changes rather slowly over time makes perfect sense, but for highly dynamic wireless networks where MNs are moving, this is an unacceptably long time. An MN would always have to wait at least 3 seconds to notice that an agent is not reachable anymore, but it may just be the case that the advertisement was simply lost. Thus, to be sure to switch to another agent, a MN has to wait even longer. Issuing solicitations is not a good solution, since they would unnecessarily flood the subnet.

Registration

When a MN recognizes that it is on a foreign network and has acquired a COA, its HA must find out about it; thus Mobile IP defines a *registration* process for this purpose. As it can be seen in Figure 4.3, the process begins when the MN, possibly with assistance of a FA, sends a registration request with the COA information. When the HA receives this request, it (typically) adds the necessary information to its forwarding table, approves the request, and sends a registration reply back to the MN. Although the HA is not required by the Mobile IP protocol to handle registration requests by updating entries in its forwarding table, doing so presents a natural implementation strategy that is widely used in practice.

Thus, registration can take place in two different ways, depending on the location of the COA:

- If the COA is at the FA, registration is done as illustrated in Figure 4.4(a). The MN sends its registration request containing the COA to the FA that forwards the request to the HA. The HA now sets up a *mobility binding* containing MN's home IP address and the current COA. Additionally, the mobility binding contains the lifetime of the registration negotiated during the registration process. Registration expires automatically after the lifetime and is deleted; therefore, a MN should register before expiration. This mechanism is necessary to avoid mobility bindings that are not used anymore. After setting up the mobility binding, the HA sends a reply message back to the FA that forwards it to the MN.

- If the COA is co-located, registration is simpler, as shown in figure 4.4(b). The MN sends the request directly to the HA and viceversa. This, by the way, is also the registration procedure for MNs returning back into their home network —they also register directly with the HA.

Tunneling and encapsulation

When a MN is registered with a HA, the HA must be able to intercept IP datagrams sent to the MN home address so that these datagrams can be forwarded via tunneling. The standard does not mandate a specific technique for this purpose, but references the Address Resolution Protocol (ARP) [123] as a possible mechanism. The HA needs to inform other nodes on the same network (the home network) that IP datagrams with a destination address of the MN in question should be delivered (at the link level) to this agent. In effect, the HA steals the identity of the MN in order to capture packets destined for that node that are transmitted across the home network.

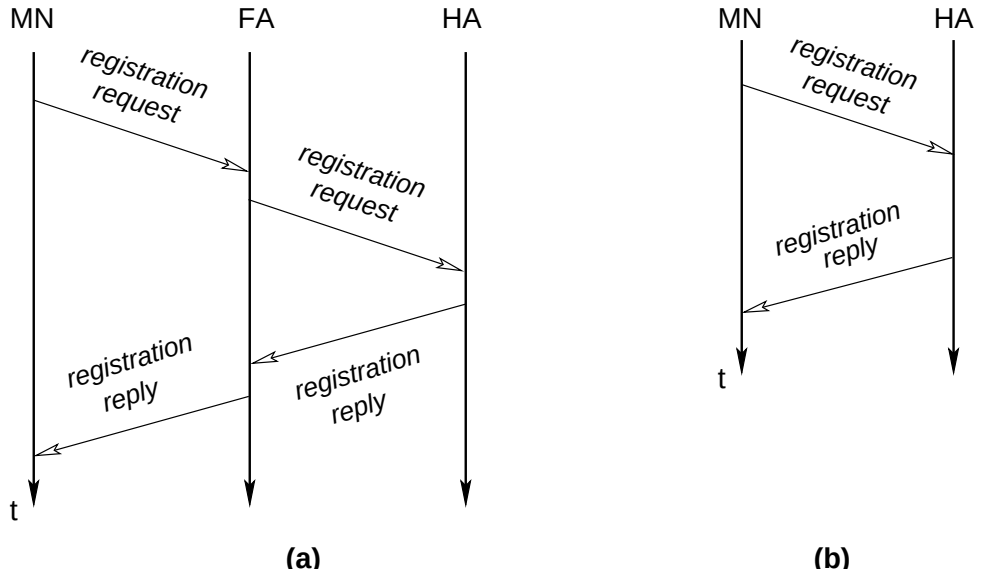

Figure 4.4: Registration of a MN (a) via the FA, and (b) direct registration with the HA.

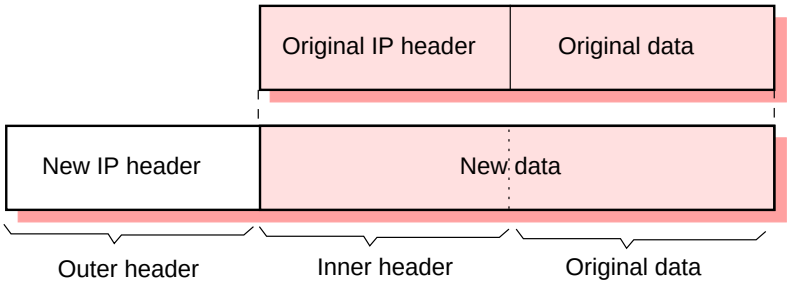

Figure 4.5: IP encapsulation.

A *tunnel* establishes a virtual pipe for data packets between a tunnel entry and a tunnel endpoint. Packets entering a tunnel are forwarded inside the tunnel and leave the tunnel unchanged. Tunneling, *i.e.*, sending a packet through a tunnel, is achieved by using encapsulation. *Encapsulation* is the mechanism of taking a packet consisting of a packet header and data and putting it into the data part of a new packet. The reverse operation, taking a packet out of the data part of another packet, is called *decapsulation*. Encapsulation and decapsulation are the operations typically performed when a packet is transferred from a higher protocol layer to a lower layer or from a lower to a higher layer, respectively. Here these functions are used within the same layer. This mechanism is shown in Figure 4.5 and describes exactly what the HA at the tunnel entry does. The HA takes the original packet with the MN as destination, puts it into the data part of a new packet and sets the new IP header in such a way that the packet is routed to the COA. The new header is also called the outer header, for obvious reasons. Additionally, there is an inner header that can be identical to the original header as is the case for IP-in-IP encapsulation encapsulation, or the inner header can be computed during encapsulation.

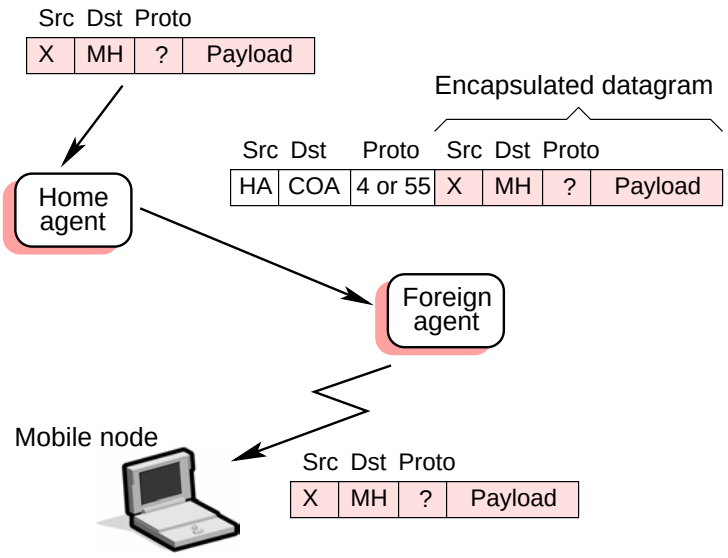

Figure 4.6: Tunneling operations in Mobile IP.

The default encapsulation mechanism that must be supported by all mobility agents using Mobile IP is *IP-within-IP* [116]. Using IP-within-IP, the HA (the tunnel source), inserts a new IP header (or tunnel header), in front of the IP header of any datagram addressed to the MN's home address. The new tunnel header uses the MN's COA as the destination IP address, or tunnel destination. The tunnel source IP address is the HA, and the tunnel header uses 4 as the higher level protocol number, indicating that the next protocol header is again an IP header. In IP-within-IP the entire original IP header is preserved as the first part of the payload of the tunnel header. Therefore, to recover the original packet, the FA merely has to eliminate the tunnel header and deliver the rest to the MN. Figure 4.6 shows that sometimes the tunnel header uses protocol number 55 as the inner header. This happens when the HA uses *minimal encapsulation* [117] instead of IP-within-IP. Processing for the minimal encapsulation header is slightly more complicated than that for IP-within-IP, because some of the information from the tunnel header is combined with the information in the inner minimal encapsulation header to reconstitute the original IP header. On the other hand, header overhead is reduced. There is also another encapsulation mechanism called *Generic Routing Encapsulation*, defined in RFC 1701 [60], that was developed prior to the development of Mobile IP.

Figure 4.7 indicates the underlying protocol support for the operations of discovery, registration, and tunneling that we have just discussed. Registration uses a transport-level protocol since it takes place between an application on the mobile node and an application in the home agent. Being a simple request/response transaction, registration does not require of all the overhead of a connection-oriented protocol such as TCP, and hence uses the simpler UDP as its transport protocol. ICMP is a connectionless protocol well suited for the discovery operation; the appropriate extensions are added to the ICMP header to implement discovery. Finally, tunneling is performed at the IP level.

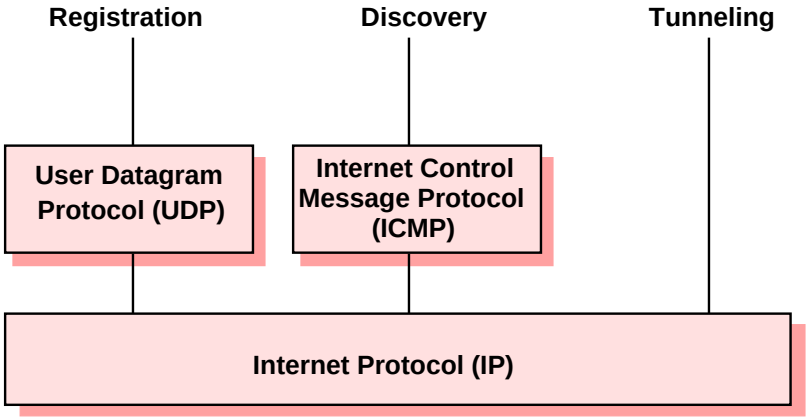

Figure 4.7: Protocols supporting Mobile IP.

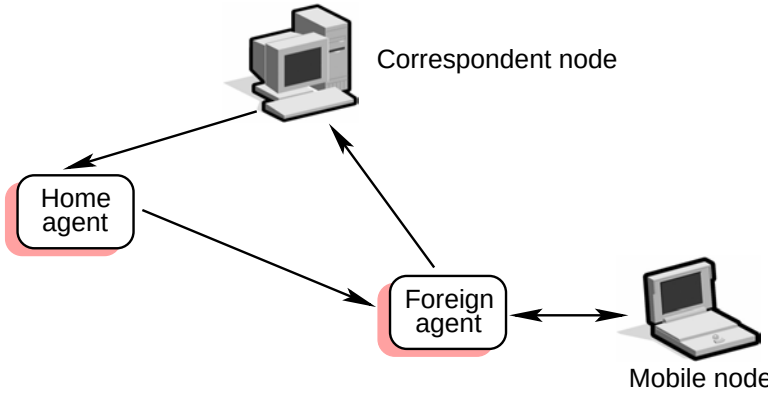

Figure 4.8: Triangular routing.

4.1.4 Mobile IP problems and optimizations

There are currently several outstanding problems facing Mobile IP, posing technical as well as practical obstacles for its deployment [31]. Some of the most notable ones are:

- **Routing inefficiencies**. In the basic Mobile IP protocol, IP packets destined to a MN that is outside its home network are routed through the HA. However, packets from the MN to the correspondent nodes are routed directly. This is known as triangle routing (illustrated in Figure 4.8). This method may be inefficient in cases such as when the correspondent host and the MN are in the same network, but not in the same home network of the MN. In such a case, the messages will experience unnecessary delay since they have to be first routed to the HA that resides in the home network. In order to alleviate this, a technique known as *route optimization* has been proposed [72]. However, implementing it requires changes in the correspondent nodes that will take a long time to deploy in IPv4. It is hoped that triangle routing will not be a problem for IPv6 mobility.

- **Security issues**. Security is a particular area of attention in Mobile IP, as MNs are often connected to the Internet via wireless links vulnerable to security attacks. For instance, during the registration procedure the HA should be convinced that it receives an authentic Registration Request from a MN and not from a bogus node. Another source of problems are firewalls, which cause difficulty for Mobile IP because they block all classes of incoming packets that do not meet specific criteria. Enterprise firewalls are typically configured to block packets from entering via the Internet that appear to emanate from internal computers. Although this permits management of internal Internet nodes without great attention to security, it presents difficulties for mobile nodes wishing to communicate with other nodes within their home enterprise networks. Such communications, originating from the MN, carry the MN's home address, and would thus be blocked by the firewall. Firewall traversal solutions have been proposed by Gupta and Glass [55], as well as by the MOIPS (Managed Objects for IP Mobility) project at BBN [161].

- **Performance and scaling issues**. Studies have shown that Mobile IP can suffer from unacceptably long handoff latencies when the mobile host is far from its home network [101]. Indeed, when the mobile is outside its home network, latencies induced by the registration process can be tolerable during low congestion periods. However, under periods of congestion there can be losses of registration messages and registration latencies reach unacceptable levels, taking as much as one second. Scalability can be a problem as the number of mobile hosts grow, but in this case the network is the bottleneck, as mobility agents (*i.e.*, HAs, FAs) can easily service at least a few hundred hosts. Suggestions have been made that using a hierarchical model to manage mobility could reduce or eliminate these performance and scaling problems [24; 56].

A lot of the problems of Mobile IP are related to the lack of features for streamlining mobility support in IPv4 [119]. Some of these problems may be solved by IPv6.

4.1.5 Mobile IPv6

While Mobile IP was originally designed for IPv4, IPv6 [40] incorporates features that support mobility much easier; several mechanisms that had to be specified separately, now come integrated with IPv6. Some of these IPv6 features include Stateless Address Autoconfiguration [145] and Neighbor Discovery [104]. IPv6 also attempts to drastically simplify the process of renumbering, which may be critical to the future of routability of the Internet [29]. Security is also a required feature for all IPv6 nodes.

Mobility support in IPv6 [120], as proposed by the Mobile IP working group, follows the design for Mobile IPv4. It retains the ideas of the home network, home agent (HA), and the use of encapsulation to deliver packets from the home network to the MN's current point of attachment. However, every IPv6 node handles address autoconfiguration, thus the mechanisms for acquiring a COA are already built into IPv6. Neighbor discovery as a mechanism mandatory for every node is also included in the specification, therefore, special FAs are no longer needed to advertise services. Combining the features of auto-

configuration and neighbor discovery means that every MN is able to create or obtain a topologically correct address for the current point of attachment.

Furthermore, every IPv6 node can send binding updates to another node, thus the MN can send its current COA to the correspondent node and HA directly. These mechanisms are an integral part of IPv6. Besides that, a soft handover is possible with IPv6. The MN sends its new COA to the old router servicing the MN at the old COA, and the old router encapsulates all incoming packets for the MN and forwards them to the new COA.

Altogether, Mobile IP in IPv6 networks requires fewer additional mechanisms from a correspondent node, MN, and HA. The FA is not needed anymore. A correspondent node only has to be able to process binding updates, *i.e.*, to create or to update an entry in the routing cache. The MN itself has to be able to decapsulate packets, to detect when it needs a new COA, and to determine when to send binding updates to the HA and correspondent node. An HA must be able to encapsulate packets.

An Alternative for IPv6 Mobility Support

Noel *et al.* have defined an architecture called LAR (for Logical Addressing and Routing) for supporting mobility using IPv6 header extensions [108]. This architecture is based on a logical addressing and routing layer on top of the network layer. The addition of the logical layer that LAR proposes does not imply adding a new layer to the protocol stack. Rather, the use of IPv6 options using header extensions has been defined.

LAR defines two types of logical addresses that are independent of the physical location of the addressed logical object (which, physically, can be a network node or a set of them) within the network:

- *Individual LAR addresses.* They are identifiers for a single node within the network. The LAR address of a node C is denoted by $@LC$.
- *Communication LAR addresses.* They are identifiers for a group of correspondents linked in a tree. These types of addresses allow point-to-point and multi-point communications.

In order to carry out LAR communications, all the machines involved have to support certain infrastructure elements such as a LAR tree, a cache, a communication controller, etc. These elements will allow to make abstractions to identify the machines in a logical way, independently of their current network address. If one or more of the machines move, the change of network address is not noticed by any of the correspondents, as they continue to communicate using LAR addresses. Mappings from logical to network addresses are carried out by the LAR infrastructure.

Hierarchical Mobile IPv6

Mobile IPv6 (MIPv6) has been developed to manage global mobility but it is not suitable for the case of micro-mobility. Recognizing this fact, some extensions have been proposed [28; 56] for MIPv6 and neighbor discovery in order to allow a Hierarchical Mobile IPv6 (HMIPv6) management model. With HMIPv6, a new entity called the Mobility

Anchor Point (MAP) is introduced. A MAP can be used at any level in a HMIPv6 network and, unlike FAs in Mobile IPv4, a MAP is not required on each subnet.

The introduction of the MAP effectively creates a hierarchical management since movement of hosts within the domain of the MAP (micro-mobility) is transparent to distant entities (such as a CN or a HA). Global mobility is handled as proposed by MIPv6. All packets for a MN pass through the MAP that will encapsulate and forward them to the MN's current address (CoA). Any changes of the local address (LCoA) implies only a new registration with the MAP, but the global address (RCoA) does not change. Handoff latency is minimized since it will take less time to bind-update a local MAP than a distant HA. Signaling is also minimized since a MN needs to perform only one local binding update to a MAP when moving between cells in a MAP domain.

As previously stated, HMIPv6 introduces extensions to the basic MIPv6 protocol (*e.g.*, the MAP), so it is possible that a HMIPv6-aware node may choose to use standard MIPv6 even if HMIPv6 capabilities are available. Furthermore, a MN using a MAP can at any time stop using it. Two methods for MAP discovery are defined. In the first method, Dynamic MAP Discovery, the MAP option is propagated from the MAP down to the MN through configured router interfaces. In the second method, Router Renumbering, the MAP option is sent from a central node to all access routers within the domain.

4.2 Micro-mobility

As several studies [83; 146] indicate, user's mobility patterns are highly localized. For instance, business professionals may spend a considerable amount of time away from their desks, but once away, most of their mobility will take place within the same building. While the mobile user is at the foreign administrative domain, there is no need to expose motion within that domain to the home agent or to correspondent hosts in other domains. Therefore, mobility management within an administrative domain should be separate from global mobility management.

In principle, Mobile IP can handle both global and local mobility. However, it requires that the mobile's home network be notified of every change in location. Moreover, route optimization extensions [72] further require that every new location be registered with hosts that are actively communicating with the mobile node. All these location updates incur communications latency and also add traffic to the wide-area portion of the internetwork. Therefore, Mobile IP does not extend well to large numbers of portable devices moving frequently between small cells. It has also been demonstrated that, when used for micro-mobility support, Mobile IP incurs disruption to user traffic during handoff, and high control overhead due to frequent notifications to the home agent [24]. Another type of protocol, a micro-mobility protocol [25], is then needed for local environments where mobile hosts change their point of attachment to the network so frequently that the basic Mobile IP tunneling mechanism introduces network overhead in terms of increased delay, packet loss and signaling.

Acknowledging the fact that Mobile IP may not be the universal end-all solution for mobility on the Internet, its performance and scalability challenges have been under dis-

cussion. Within this context, the Mobile IP working group has recently started discussing the subject of micro-mobility protocols. There are several attributes that micro-mobility protocols aim for:

- *Minimum (or zero) packet loss.* Fast handoff techniques have been developed to achieve this, and they may also *reduce latency or delay.*
- *Reduced signaling.* Techniques for locating mobile hosts, known as paging, have been proposed in order to reduce signaling. *Reduced registration* is also an outcome of these techniques.

Another important aspect that has received little attention in the design of micro-mobility protocols is that of quality of service (QoS). Triangular routing, address translation, and complex interaction between agents make Mobile IP unsuitable for QoS support in local environments [30; 63; 101]. Work in progress within the IETF's SeaMoby working group is now addressing problems related to QoS in mobile environments, although not particularly for the case of micro-mobility.

4.2.1 Hawaii

Ramjee *et al.* [127] have presented HAWAII (Handoff-Aware Wireless Access Internet Infrastructure) as an alternative for providing domain-based mobility (a.k.a. micro-mobility). Their work has also been submitted for consideration to the IETF, but hasn't advanced beyond the level of Internet Draft [126]. Under their approach, Mobile IP is used as the basis for mobility management in wide-area wireless networks, but new methods for managing mobility within an administrative domain are developed. One point worth highlighting is that mobile hosts retain their network address while moving within a domain; this way, the Home Agent (HA) —if using Mobile IP— and any corresponding hosts, are not aware that the host has performed intra-domain mobility.

Dividing the network into hierarchies, loosely modeling the autonomous system hierarchy used in the Internet, is part of the HAWAII approach. Indeed, the gateway into each domain is called the *domain root router*, and each host is assumed to have an IP address and a home domain. As already stated, hosts retain their address while moving within a domain, so, when packets destined to a mobile host arrive at the domain root router, they are forwarded over specially established paths to reach the mobile host. However, if the mobile host moves to a foreign domain, traditional Mobile IP mechanisms are used. This means the use of co-located care-of addresses and packet forwarding via tunnels by home agents. If the foreign domain is HAWAII-compliant (*i.e.*, nodes have been set-up to handle their protocols), the mobile obtain its new (care-of) address from the foreign domain, and will keep this same address for moving within that domain, analogously to what would happen in its home domain.

When a mobile host moves within a domain (performing a handoff), all involved forwarding tables are modified to redirect packets to the mobile's new location. These changes are made under one of four possible *path setup* schemes that determine when,

how, and which routers are updated, namely: MSF, SSF, UNF, and MNF. These are grouped into forwarding and non-forwarding schemes:

- *Forwarding schemes.* Packets are first forwarded from the old base station to the new base station before they are diverted at the cross-over router. MSF and SSF are forwarding path setup schemes.
- *Non-forwarding schemes.* Packets are diverted at the cross-over router to the new base station, resulting in no forwarding of packets from the old base station. UNF and MNF are non-forwarding path setup schemes.

Under MSF, due to the order in which forwarding tables are modified, transient routing loops can occur. The possibility also exists of having mis-ordered packet streams, due to the particular forwarding method. In order to overcome the problems of MSF, another forwarding scheme is proposed (SSF), which takes into account not only the source IP address and forwarding interface, but also the source interface(s). This poses a greater deal of complexity but helps to solve the mis-ordered streams and transient loop problems of MSF. The authors show that this added complexity may not be worthwhile because typical handoffs involve routers that are only one or two hops away.

The order and nature of the redirection under both UNF and MNF assures that no packets are lost or mis-ordered, and that no transient loops occur. Under MNF there is a short moment when the crossover router performs a sort of dual-cast sending data to two outgoing interfaces. UNF is meant to be used with networks where the mobile host is able to listen/transmit to both the old and new base stations simultaneously for a short duration, and MNF for the case when the mobile can only listen/transmit to one base station. These schemes are less complex and more likely to be worth implementing[2], so we will cite an example given on their paper [127], in order to detail how they work.

Under UNF, depicted in Figure 4.9(a), when the new base station receives the path setup message, it adds a forwarding entry for the mobile host's IP address with the outgoing interface set to the interface on which it received the message. It then performs a routing table lookup for the old base station and determines the next hop router, Router 2. The new base station then forwards Message 2 to Router 2. This router performs similar actions and forwards Message 3 to Router 0. At Router 0, the cross-over router in this case, forwarding entries are added such that new packets are diverted directly to the mobile host at the new base station. Eventually, Message 5 reaches the old base station that then changes its forwarding entry and sends an acknowledgment, Message 6, back to the mobile host.

The MNF scheme is very similar to the UNF scheme. The main difference is that the cross-over router, Router 0, multicasts data packets for a short duration. In Figure 4.9(b), Router 0 dual-casts data packets from interface A to both the new and old base stations after it receives Message 3 and until it receives Message 6. This helps limit packet loss in networks in which the mobile host can only listen to a single base station.

The only claims for QoS support in HAWAII are basically based on two facts:

[2]Hawaii's mobility management solutions have only been validated by simulations and no known implementation exist.

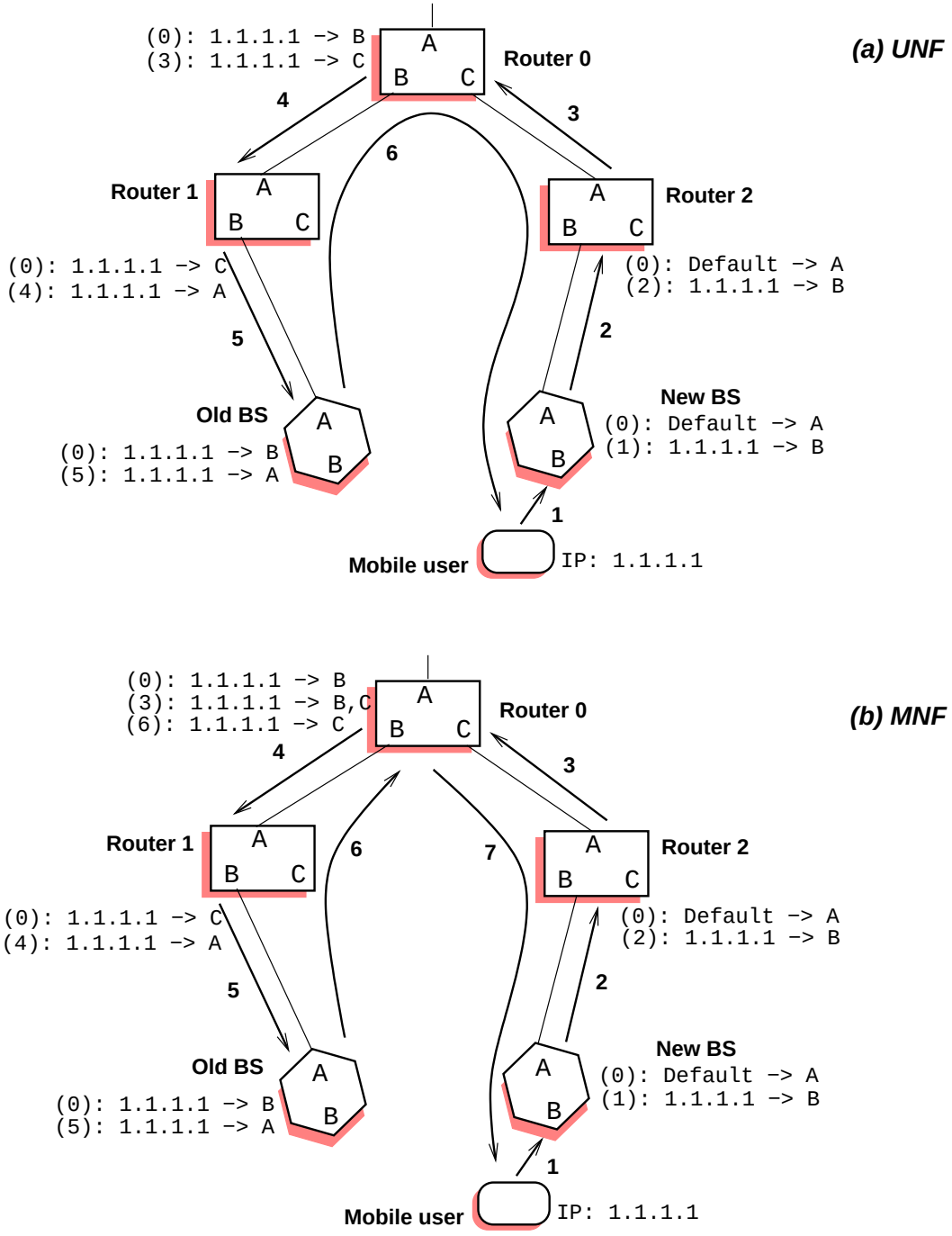

Figure 4.9: Hawaii's path setup using non-forwarding schemes.

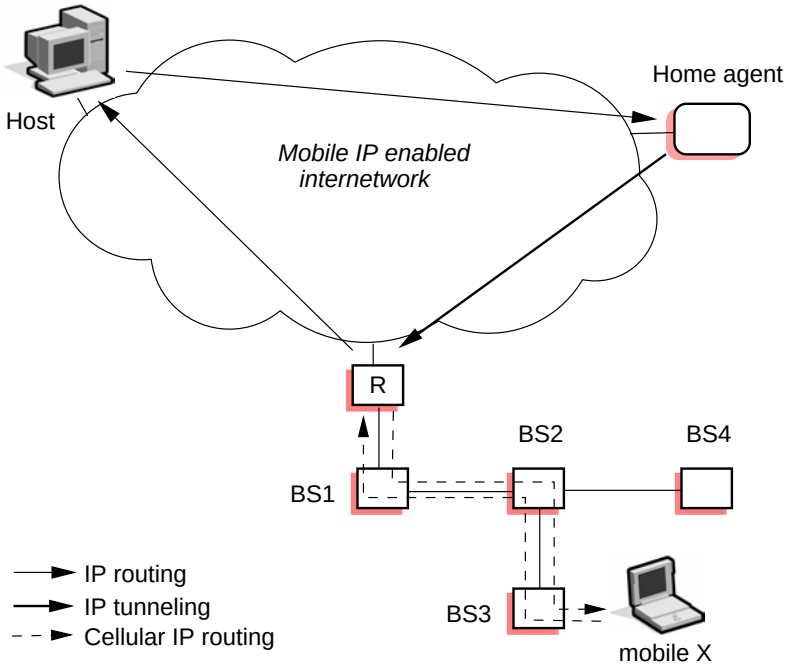

Figure 4.10: Cellular IP access network.

1. the low latencies of handoffs and redirections
2. the support (or at least non-interference with) other QoS techniques and protocols, such as RSVP.

4.2.2 Cellular IP

Being aware that mobility protocols such as Mobile IP are not suitable for the specific problems of local mobility, the Cellular IP project[3] was started to propose a micro-mobility architecture [151; 26]. Cellular IP aims to integrate cellular technology principles with the IP networking paradigm; this poses hard challenges, as there are fundamental architectural differences between cellular and IP networks. A Cellular IP node constitutes the universal component of a Cellular IP network, since it serves as a wireless access point but at the same time routes IP packets and integrates cellular control functionality traditionally found in Mobile Switching Centers (MSC) and Base Station Controllers (BSC). Cellular IP nodes are modified IP nodes where standard routing is replaced by Cellular IP's own routing and location management functions.

A Cellular IP network is connected to the Internet via a gateway router. Mobility between gateways (*i.e.*, Cellular IP access networks) is managed by Mobile IP, while mobility within access networks is handled by Cellular IP. Mobile hosts attached to the network use the IP address of the gateway as their Mobile IP care-of address. Figure 4.10 illustrates the path of the packets addressed to a mobile host, where plain Mobile IP

[3] http://comet.columbia.edu/cellularip

(without route optimization) is used. Packets will be first routed to the host's home agent and then tunneled to the gateway. The gateway "de-tunnels" packets and forwards them toward base stations. Inside the Cellular IP network, mobile hosts are identified by their home addresses and data packets are routed without tunneling or address conversion. Packets transmitted by mobile hosts are first routed to the gateway and from there on to the Internet.

An evaluation of Cellular IP's performance is given in [152]. The fact that Cellular IP relies on the installation of specialized nodes in a domain, and that the gateway is critical to the reliability of the whole domain, can certainly be a drawback. Also, the current protocol supports best effort traffic only [26].

As already mentioned, Cellular IP nodes integrate location management and routing functions; so, what follows is an overview of such functionalities, including routing, handoff and paging.

Routing

In a Cellular IP network, none of the nodes know the exact location of a mobile host. Packets addressed to a mobile host are routed to its current base station on a hop-by-hop basis where each node only needs to know the outgoing port to forward packets on. The gateway periodically broadcasts a beacon packet that is flooded in the access network. Base stations record the interface from which they received the beacon and use it to route packets toward the gateway. All packets transmitted by mobile hosts, regardless of their destination address, are routed to the gateway using these routes.

The way Cellular IP routing works can be exemplified using the scenario in Figure 4.10: packets are transmitted by a mobile host with IP address `X` and enter `BS2` through its interface `a`. In the routing cache of `BS2` this is indicated by a mapping `(X, a)`. This mapping remains valid for a system specific time *route-timeout* and its validity is renewed by each data packet that traverses the same interface coming from the same mobile. As long as the mobile host is regularly sending data packets, base stations along the path between the mobile host's actual location and the gateway maintain valid entries in their routing cache forming a soft-state route between the mobile host and gateway nodes. Packets addressed to the same mobile host are routed on a hop-by-hop basis using the established routing cache. If the mobile host is not regularly transmitting data packets, it can however maintain its routing cache mapping by sending empty data packets, called *route-update* packets, at regular intervals.

Handoff

All handoffs in Cellular IP are initiated by the mobile host. The *hard handoff* algorithm is based on a simplistic approach to mobility management that supports fast and simple handoffs at the price of potentially some packet loss. Hosts listen to beacons transmitted by base stations and initiate handoff based on signal strength measurements. As the host approaches a new base station, it redirects its data packets from the old to the new base station. The first of the redirected packets will automatically configure a path of routing cache mappings for the host, this time to the new base station. Handoff latency is the time that elapses between the handoff and the arrival of the fist packet through the new route. During this time, downlink packets may be lost. The mappings associated with

the old base stations are not cleared at handoff, rather, they timeout as the associated soft-state timers expire.

There is a time equal to the timeout of the route cache mapping, during which packets addressed to the mobile host will be delivered at both the old and new base stations. This way, if the host's radio device is capable of listening to both base stations for a short time, a semi-soft *semi-soft handoff* will take place, which provides probabilistic guarantees instead of fully eliminating packet loss.

Paging

Idle mobile hosts periodically generate short control packets, called *paging-update* packets, sending them to the nearest available base station. Similar to data and route-update packets, paging-update packets are routed on a hop-by-hop basis to the gateway. Base stations may optionally maintain a paging cache, which is updated by any packet sent by mobile hosts, including paging-update packets. Paging is thus a mechanism mean to handle location management of idle mobile hosts.

4.3 Seamless Mobility: SeaMoby

The Transport Area of the IETF has recently formed a working group focused on context transfer, handoff candidate discovery, and seamless mobility, known in short as SeaMoby[4]. As a recently created group, most of their activities are still considered work in progress, so most of the cited references (mainly Internet Drafts) should be considered as work in progress descriptions, and not as permanent reference material. It is expected that the working group, along with other groups within the IETF, and with input from standards defining organizations and industry groups, will produce a protocol —or a set of protocols— that will enable the provision of real-time services over an IP infrastructure, that will also work with minimal disruption across heterogeneous wireless, and wired, technologies. The working group will also ensure that their solutions are compatible with Mobile IP.

One of the main motivations for the constitution of the SeaMoby working group was that, during discussions of the Mobile IP working group concerning fast handoffs, the need for a new protocol allowing state information to be transferred between edge mobility devices was identified. This is called context transfer, and some examples of information that could be used to transfer include authorization, authentication and accounting (AAA) information, security context, QoS properties assigned to the user, etc. Context transfer aims for seamless mobility; this requires preserving the capability, security and quality of service offered to a mobile node during handovers [18].

The purpose of context transfer is to sustain the services being provided to a mobile node's traffic during handover, and the working group supposes that this enhancement to mobility solutions will ultimately result in an improvement in handover performance [97; 142]. Providing context transfers for seamless mobility was one of the original goals of SeaMoby, but the working group later identified two more technologies important for

[4] `http://www.ietf.org/html.charters/seamoby-charter.html`

their work: handoff candidate discovery and dormant mode host alert (a.k.a. IP paging).

- **Handoff candidate discovery**. Fast handoff solutions, such as those being developed by the Mobile IP working group [43], assume that a set of candidate nodes for handoff has been chosen by means of some mechanism, which the working group is not developing. The Seamoby working group has also identified the right selection of candidate nodes as a pre-condition for seamless mobility. Furthermore, the group has recently identified some issues concerning the requirements and protocol needed for handoff candidate discovery, and published a problem statement [36].
- **Dormant mode host alerting**. Also known as *IP paging*. Mobile nodes often support a state (called dormant mode) in which the mobile restricts its ability to receive normal IP traffic by reducing monitoring of radio channels. IP paging enable network devices to track a mobile that has moved from its last point of attachment, while in dormant mode, allowing the mobile's packets to be delivered. This work within the SeaMoby working group is also still in the phase of problem statement [80; 79].

As of today, the SeaMoby working group has scheduled work and milestones up until june 2002. During that period of time they will clearly define all the requirements for solving the problems stated above, and develop the corresponding protocols. They have stated that all work produced within the group will support IPv4 and IPv6, will follow the congestion control principles in RFC 2914 [49], and will undergo a security review prior to the work's completion.

4.4 Mobility with QoS

The proposals we have presented up to now deal with mobility without taking into account the problem of providing QoS guarantees to mobile nodes. Researchers have observed that mobility and QoS are problems that have a strong relation. Therefore, some recent proposals deal with these problems in an integrated way.

4.4.1 In-band signaling for QoS: Insignia

Before presenting its main features, it is worth highlighting that the type of mobile networks that Insignia [90] deals with is of a different nature to what we are interested in our work. Insignia deals with Mobile Ad-hoc Networks (MANETs). These are autonomous distributed systems that comprise a number of mobile nodes connected by wireless links, forming arbitrary time-varying wireless network topologies [33]. MANETs rise very particular problems that can not be solved by solutions proposed for wireless access networks with a fixed wired infrastructure support (*e.g.*, fixed routers). For this reason, the IETF has formed a working group[5] to address issues related to MANETs.

[5] `http://www.ietf.org/html.charters/manet-charter.html`

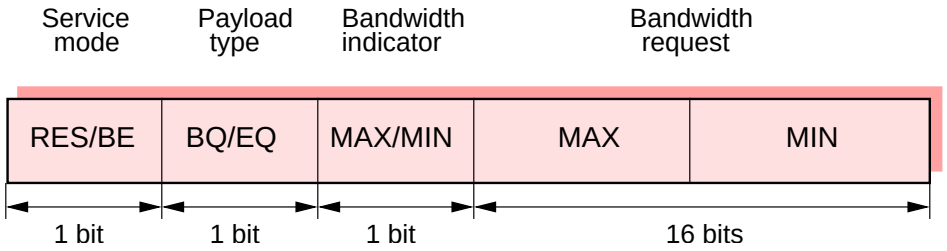

Figure 4.11: The Insignia IP option.

Our interest in Insignia is thus oriented toward its use of in-band signaling in IP networks, in order to solve problems related to mobility and QoS support. The term *in-band signaling* refers to the fact that the control information is carried along with data. In contrast, the term *out-of-band signaling* refers to the fact that the control information is typically carried in separate control packets and on channels that may be distinct from the data path.

The Insignia signaling system supports several protocol commands that drive fast reservation, fast restoration, and end-to-end adaptation mechanisms. These commands are encoded using the IP Option header field (as seen in Figure 4.11). We will continue with an overview of them. More detailed descriptions can be found on a proposal submitted to the IETF [91] or in a paper by Lee and Campbell [89]. Results of evaluations by simulation have also been published [92; 88].

Fast Reservation

A packet carrying a reservation request has the service mode of its Insignia IP Option set to Reservation (`RES`) mode, the payload type set to either base QoS (`BQ`) or enhanced QoS (`EQ`), and the bandwidth indicator to minimum/maximum (`MIN/MAX`), and valid bandwidth requirements. Reservation packets traverse intermediate nodes executing admission control modules, allocating resources and establishing flow-state at all intermediate nodes between source-destination pairs. A source node continues to send reservation packets until the destination node completes the reservation set-up phase by informing the source node of the status of the flow establishment phase using a QoS reporting mechanism. When a reservation packet is received at the destination node, the status of the reservation phase is determined by inspecting the service mode bit in the IP Option field. The service mode bit could be set to `RES` for reservation or `BE` (Best Effort) for no reservation. If the bandwidth indicator bit of the Insignia IP option is set to `MAX`, it implies that all nodes between a source-destination pair have successfully allocated resources to meet the base and enhanced bandwidth requirements in support of the max-reserved service. On the other hand, if the bandwidth indicator is set to `MIN`, this means that only the base QoS can be currently supported (*i.e.*, min-reserved mode). In this case, all reservation packets with a payload of `EQ` received at the destination will have their service level flipped from `RES` to `BE` by the bottleneck node.

Insignia is designed to operate over unidirectional and bidirectional links. However, reservations are only established on the forward link between source and destination nodes. The reception of a QoS report allows a source node to remove any partial reservation

between the source and bottleneck nodes by sending `EQ` packets in the `BE` service mode. In this case, any resources reserved for `EQ` packets between the source and bottleneck nodes are automatically released by the Insignia soft-state resource management mechanism active at all intermediate routers.

Fast Restoration

Reservation-based flows are often re-routed during the lifetime of on-going sessions due to host mobility. The goal of flow restoration is to re-establish reservation as quickly and efficiently as possible. Rerouting active flows involves a routing protocol [112; 122; 73] (to determine a new route), admission control and resources reservation for nodes that belong to a new path. Fast restoration procedures also call for the removal of flow-state at nodes along the old path. In an ideal scenario, the restoration of a flow can be accomplished within the duration of a few consecutive packets given that an alternative route is cached; this type of restoration is called *immediate restoration*. If no alternative route is cached the performance of the restoration algorithm depends on the speed at which the routing protocols can discover a new path. When an adaptive flow is re-routed to a node where resources are unavailable, the flow is degraded to the best effort service. Subsequently, downstream nodes receiving these packets with degraded service do not attempt to allocate resources or refresh the reservation state associated with a flow. In this case, the state associated with a flow automatically times out and resources are de-allocated. A reservation may be restored if resources are freed up at a bottleneck node or further re-routing of flows allow the restoration process to complete; this is called *degraded restoration*.

If a flow remains degraded for the duration of its session, it is deemed *permanently degraded*. The enhanced QoS components of an adaptive flow may be degraded to the best effort service (*i.e.*, min-reserved mode) during the flow restoration process if the nodes along the new path can only support the minimum bandwidth requirement. Insignia supports adaptive soft-state timer control by means of a reservation system that "tunes" the duration of individual reservation timers. Reservation-based schemes built on a soft-state resource management approach are very suitable for highly mobile environments. They argue [89] that an adaptive soft-state timer approach resolves a number of pathologies found in reservation-based MANETs, such as false reservation and resource lockup, which limit performance.

End-to-end Adaptation

The Insignia QoS framework actively monitors network dynamics and adapts flows in response to observed changes based on a user-supplied adaptation policy. Reception quality of a flow is monitored at the destination node and based on an application-specific adaptation policy, actions may be taken to adapt the flow to observed network conditions. Actions taken are conditional on the adaptation-policy resident at the destination node, *e.g.*, adaptation policy may choose to maintain the service level under degraded conditions. Other policy could scale-up flows whenever resources become available. The application is free to program its own adaptation policy that is executed by Insignia through interaction between the destination and source nodes. Liao [98] describes the adaptation policy API with great detail. The Insignia signaling system supports three adaptation commands that are sent from the destination host to the source using QoS reports:

- A scale-down command requests a source node to send its enhanced QoS packets as best effort or its enhanced QoS and base QoS as best effort.
- A drop command requests a source node to drop its enhanced QoS packets or enhanced and base QoS packets (where the term "drop" means the source stops transmitting these packets).
- A scale-up command requests a source node to initiate a reservation for its base and/or enhanced service quality.

Insignia does not embed application-specific adaptation policy in the network (*e.g.*, adaptation timescales, actions). Rather, it provides a simple adaptive reservation-based service model that supports service differentiation between `BQ` and `EQ` packets. Applications are free to map this service differentiation to data as they wish, monitor the network, and adapt to resource availability (by monitoring the bandwidth indicator bit) over the timescales the application considers appropriate. In essence, Insignia provides a simple API to the network to implement sophisticated adaptation policies at the edge (*i.e.*, source/destination) in a scalable manner.

4.4.2 Integrated services: CLEP and MIR

The IntServ model has been proposed as the basis for QoS provision over local networks supporting mobile hosts [143; 66; 94]. Recently, an architecture was proposed [93] that uses a signaling protocol for QoS reservations, much in the spirit of IntServ. In that architecture, a protocol called Controlled-Load Ethernet Protocol (CLEP) [21] serves as the basis for their Mobile IP Reservation (MIR) protocol [95].

CLEP is an implementation over Ethernet of the Controlled Load service defined by Wroclawski [158]. It provides flows with a quality of service similar to what they would receive on an unloaded network. In order to obtain this service, access controllers built around token bucket filters are placed on the outgoing interfaces of the nodes. This form of admission control lets packets pass only if there is enough bandwidth for them. The parameters of the token buckets filters in network elements are managed dynamically using a distributed protocol. CLEP provides the shared medium with the following properties [6]:

- A steady bandwidth in overload condition,
- a guaranteed bandwidth for streams which have a reservation,
- a fair share of bandwidth for best effort streams,
- and isolation of the streams that have QoS requirements.

MIR extends CLEP to mobile environments based on a model that uses Mobile IP and IEEE 802.11. This way, each 802.11 cell is seen as an IP domain where the base station is also a Mobile IP foreign agent. MIR is a distributed protocol that allows

each cell to be managed separately, depending on the local mobility of nodes within the cell. An important difference between CLEP and MIR is that the former provides hard bandwidth guarantees, while the latter, due to the fact that is meant for wireless mobile environments, can only give statistical guarantees.

When the load in a cell is less than a threshold $S1$, new flows and privileged handoffs are accepted. However, if the load exceeds $S1$, new flows are not accepted. When a mobile hosts receives and advertisement from a foreign agent, it then asks the agent to make a passive reservation for privileged flows. This reservation becomes active when the hosts executes a handoff using Mobile IP. After the handoff the reservation is actually used to send data. It may be the case that a passive reservation was made up to a higher threshold $S2$, but that at the moment of actually sending data not all of it is used. Thus, a passive reservation may exceed the active reservation.

The authors have stated that MIR allows the provision of QoS guarantees at the local level. However, mainly due to scalability problems, providing end-to-end guarantees across the Internet is not possible by using only MIR [95]. They have envisaged [93] the use of MIR at the edges (locally within individual cells), complemented by a Diffserv approach for the core network. Also, a variation of CLEP based on the Diffserv model —called CLEP-DS— has been proposed [6].

4.4.3 Wireless ATM

The proposal for QoS in mobile networks that we will review now is, in many ways, different from the others we have presented. Notably, it is not IP-based (although TCP/IP protocols could be made to co-exist with it). Equally important, it is a proposal that comprises several layers, not only the networking layer.

Because of its modern built-in features, such as QoS support, ATM [59] has been considered as an obvious candidate for providing mobility and QoS functions over wireless networks. Thus, wireless ATM (WATM) aims at extending ATM services to wireless environments. Similar to the ATM technology for fixed networks, WATM does not only describe a transmission technology, but tries to specify a complete communication system. For example, 802.11 covers only local area networks, Bluetooth deals only with piconets, and Mobile IP only works on the network layer. In contrast, WATM tries to build up a comprehensive system covering physical layer, media access, routing, integration with the fixed ATM network, service integration with ISDN, etc. Although active work is being done within the ATM Forum[6], the industry, and the academic world, there is currently no standard covering all aspects that WATM deals with. In this brief review we will present only the aspects that are the most relevant to our work.

Integrating ATM features in wireless environments is not a trivial task. An important problem arises from the fact that ATM was designed for media whose bit error rates are very low (about 10^{-10}). Wireless networks, which are typically noisy, do not offer such low bit error rates. Also, WATM was designed for bandwidth-rich environments, but the situation in most wireless networks is bandwidth scarcity on a shared medium [7].

[6] `http://www.atmforum.com`

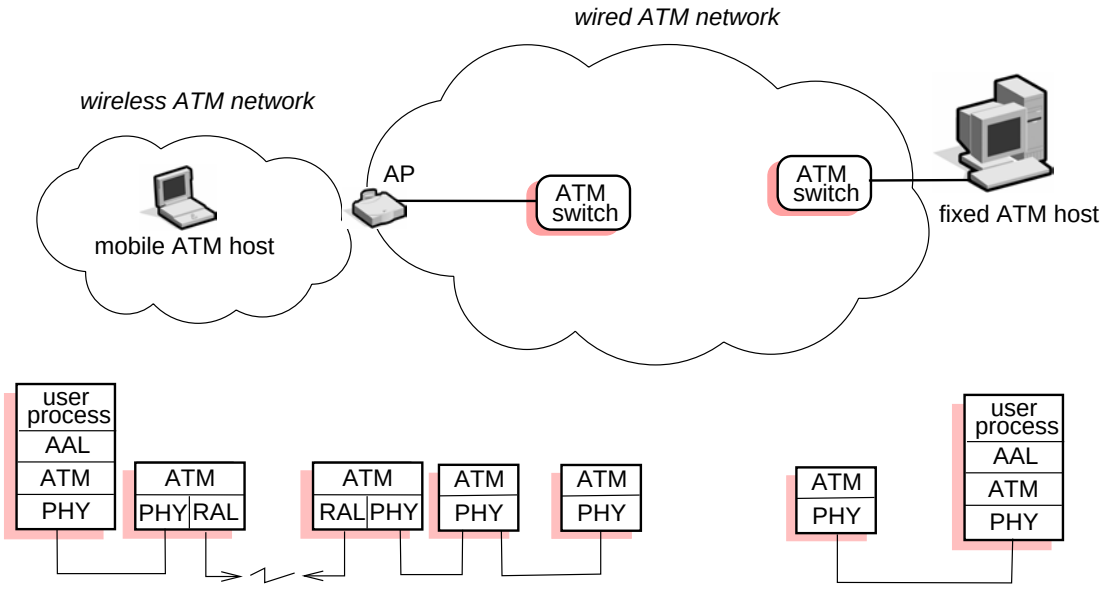

Figure 4.12: A wireless ATM system and the corresponding protocol stacks.

Figure 4.12 depicts a scenario where a mobile WATM-enabled node connects to a fixed ATM network via an access point (also WATM-enabled). As the corresponding protocol stacks show, higher protocol layers remain the same for wireless and wired transmission, so users processes should not be aware if wireless access is used or not. However, data transfer over the radio link might be completely different from transfer over wires. The radio access layer (RAL) can perform header compression to increase efficiency, apply FEC for a higher reliability, or insert new control information in an ATM cell header. The RAL includes LLC, MAC and PHY layers. It should be noted that on the user side of the terminal adapter and the network side of the access point, there appears to be no difference to the usual ATM cell stream.

MAC functions within the RAL include controlling the simultaneous access to the medium, *i.e.*, defining a MAC protocol, a PDU format, and a MAC algorithm. In addition to this, it is necessary to provide interfaces to the physical and logical link control layers, as well as support for user mobility. Schiller [130] also defines the following general goals:

- *Logical channels.* As already seen (figure 4.12), the MAC layer within the RAL connects the ATM layers of the access point and the mobile host. Thus, the MAC layer must also support logical channels for ATM virtual connections, each associated with a set of ATM QoS and traffic parameters.
- *QoS/traffic parameters.* One of the advantages of ATM over other best effort technologies (*e.g.*, Mobile IP on 802.11) is that it can guarantee end-to-end QoS. The MAC in WATM should not compromise this guarantees, and should support the QoS and traffic parameters defined in the ATM specification.
- *Architecture.* The WATM architecture uses an infrastructure network, so it is mandatory for the MAC layer to support such architecture. However, support for

ad hoc configurations is optional.

- *Service provision.* Several minimum requirements are defined regarding the provision of different ATM services by the MAC layer. For instance, the CBR, VBR-rt, VBR-nrt, ABR, and UBR service classes —including QoS control— should be supported. The minimum efficiency of the MAC layer should be 60-75%. The peak data rate should be at least of 25 Mbps with a sustained data rate of 6 Mbps.

Another important aspect under development is mobility management, which includes location management, connection management, and handoff management. Location management deals with the problems of determining the current position of a mobile host, providing the host with a permanent address, and ensuring security features such as privacy, authentication and authorization. Recall that ATM is connection-oriented, so a serious problem arises in mobile networks as endpoints move, and segments of connections need to be torn down or reestablished. To address this problem, the BAHAMA project [45] proposes emulating connectionless networks using ATM technologies. This is just one approach, but others suggest keeping the connection-oriented nature of ATM [147; 14] and list the following requirements for handoffs:

- *Handoff of multiple connections.* If a mobile host has several connections established, all connections should be re-routed to the new location during handoff. If there are not enough resources available to support all connections, it could be possible to accept QoS degradation, or simply dropping single connections.

- *Handoff of point-to-multipoint connections.* ATM supports point-to-multipoint connections seamlessly, and it is required for WATM to also do so. The complexity of the scheme may require to contemplate some restrictions, though.

- *QoS support.* QoS configurations should be preserved when performing handoff from cell to cell. However, if the new cell does not permit this, there should be functions for QoS re-negotiation, or possibly dropping some connections according to their priority.

- *Data integrity and security.* Cell loss should be minimized and cell duplication or re-ordering should be totally avoided. Security levels should also be kept after handoff.

- *Signaling and routing support.* Support should be present to identify mobility-enabled switches in the network, to determine the presence of adjacent radio switches (access points), and to re-route partial connections in the handoff domain.

- *Performance and complexity.* WATM systems are inherently complex due to their support of functions like QoS-enabled connections. Thus, it is necessary that mobility functions (*e.g.*, handoff) be simple. It should also be considered that, for performance reasons, ATM switches are mostly hardware-based, so it is difficult to upgrade them and integrate new features such as mobility into them. In the mobile hosts, it is also desirable that the code for mobility management be simple.

As mentioned before, since ATM has been designed to provide QoS guarantees, one of the main expected advantages of WATM over other mobility alternatives (*e.g.*, Mobile IP) is that it will also support QoS. Maintaining QoS over time, even in the presence of handoffs, is a great challenge. The following handoff scenarios can be envisaged [130]:

- *Hard handoff QoS.* If a certain QoS level was given in the current cell due to the availability of resources, nothing can guarantee that enough resources will be available in the new cell at the moment of handoff. If the mobile hosts and its applications can not adapt to a changing situation (*e.g.*, lower QoS levels), then the connection could be cut off or handoff denied.
- *Soft handoff QoS.* A more flexible approach consists of giving only statistical QoS guarantees —even in the current cell— and having applications adapt after handoff. For instance, if the new cell is low in resources, either the host closes some of its connections to free resources, or tries to adapt (degrade) the QoS of one or more connections.

4.5 Chapter Conclusions

The Internet was born in an era when no mobile networking equipment was available. Therefore, all the basic protocols were designed under the tacit assumption that the end-points of a communication would stay fixed all along. With the arrival of modern communications equipment that allow these end-points to change their position, new protocols for handling mobility were proposed. Mobile IP is the culmination of many of these propositions. It has been showed that IP mobility is essentially an address translation problem, and that it can be solved at the network layer.

While a mobile host moves within an administrative domain, there is no need to expose motion to outside correspondent hosts. Mobility within the same domain, named micro-mobility, has very particular characteristics that make inappropriate to handle it with global mobility protocols like Mobile IP. With a micro-mobility protocol, moving nodes do not acquire new addresses. Some general goals of micro-mobility protocols are to minimize latency and packet loss, and to reduce signaling. Hawaii and Cellular IP are some of the most relevant micro-mobility protocols.

Mobility and QoS have been mostly treated as non-related problems, with independent solutions for each. However, more recently, some proposals have coupled solutions for both problems. Insignia uses the concept of in-line signaling for making QoS reservations in mobile ad hoc networks, where all nodes —including routers— have moving capabilities. The IETF has issued a couple of models for QoS provision in IP networks and one of them, the IntServ model, has been applied to mobile networks with proposals such as CLEP and MIR. Current work within the IETF deals with the problems of seamless mobility in order to sustain services offered to a mobile node after handover.

None of the presented solutions satisfy our goals for integrated mobility and QoS management. As thoroughly discussed, Mobile IP is not suitable for micro-mobility environments. Solutions like Hawaii and Cellular IP handle micro-mobility but do not

integrate QoS support. Insignia is meant for ad hoc networks where all network nodes (including routers) move and the topology of the network changes dynamically. Seamless mobility is mostly at the early stage of problem definition and no complete or definitive solutions have been proposed. CLEP and MIR propose the integration of mobility and QoS management in IP networks but, as we will discuss later, we do not consider the IntServ model to be appropriate for the characteristics of WLANs.

Integrated Solutions for QoS and Mobility Management

Chapter 5

Integrated Solutions for QoS and Mobility Management

> *The concept is interesting and well-formed, but in order to earn better than a 'C', the idea must be feasible.*
>
> *— A Yale University management professor in response to student Fred Smith's paper proposing reliable overnight delivery service (Smith went on to found Federal Express Corp.)*

We have seen how several proposals for mobility management and QoS management in the context of wireless local are networks have been independently developed (*cf.* Chapters 3 and 4). Most of these proposals address either one problem or the other, but not both. For instance, Hawaii [127] is a proposal for mobility management, that leaves the problem of QoS to the eventual integration of protocols such as RSVP. We argue that:

- Mobility and QoS management problems in wireless networks are tightly coupled, and solutions to address these problems should take into account that fact.
- As a result, micro-mobility solutions should not only strive to minimize latency and packet loss, but also aim for the smooth integration of QoS management solutions.
- Also, in the case of IP-based local wireless networks, complex signaling solutions (such as those proposed by IntServ) imply too much overhead that is contrary to the dynamic nature of this type of networks. Simpler solutions (such as those proposed by DiffServ) are more adequate for these environments.

With this in mind, we proceed to detail our propositions all along the current chapter. In order to validate our ideas and give proofs of their feasibility, we have developed prototypes of our architectures and protocols, and have made (empirical) tests and experiments. These proofs of concepts will be presented in the next chapter.

5.1 Evaluation of our Wireless Network

Before describing our proposals for QoS and mobility management, we will present some empirical tests we carried out to determine the performance of 802.11-based WLANs. Our work does not deal with proposing mechanisms for QoS at the MAC level. However, it is important to determine the actual performance features of the WLAN we will use in order to be realistic when proposing QoS mechanisms on top of such WLAN.

Stallings [138] compares 802.11 to six other (wired) LAN technologies and shows that it is the worst regarding QoS support. According to his analysis, some of the technologies reviewed are able to handle up to eight different priorities, while in 802.11 priority is not supported. Furthermore, the CSMA/CA mechanism tries to give fair access to the medium to all stations, countering the need for priorities when trying to provide QoS. An optional point coordination function (PCF) provides a time-bounded service by means of using an access point as coordinator that allows priority access to the medium to only one station at a time. Such function introduces further overhead [85] that is detrimental for QoS and does not work in the ad hoc mode. All this goes without mentioning problems that are inherent to the wireless medium, such as interference, attenuation, and fading.

Some efforts are under way to provide QoS using the 802.11 MAC. Aad and Castelluccia [2] have proposed three mechanisms for service differentiation at the MAC level. The first one consists on variation of the contention window according to flow or user priorities; the second one is based on variation of the DIFS, and the third one assigns different maximum frame sizes to different priorities. The 802.11e working group within the IEEE is also currently working on providing QoS using a variation of the basic 802.11 standard. Bianchi and Campbell [16] take another approach: instead of proposing new MAC protocols or modifying existing ones, they propose a programmable MAC. Indeed, they describe a programmable middleware platform that allows applications to program service specific requirements over the radio hardware. In their programmable framework there is a MAC control plane that supports very elementary service classes, characterized by simple QoS descriptors, and new services and QoS models can be built on top of these.

5.1.1 Test environment

All experiments were carried out at LSR's facilities, which are in a relatively old building that, as will be shown ahead, is far from being an ideal test environment. The building is compartmentalized with walls of different widths and materials, ranging from 10 cm concrete walls, to heavy concrete 15 cm walls. There are also some support columns around the stairs and in other parts of the building. The length of the building is 75 m by 12 m width. Figures 5.1 and 5.2 show a simplified blueprint of the building.

The following equipment was used for the performance evaluations:

- Lucent Wavelan IEEE 802.11 cards, which allow for (nominal) throughputs of 4 and 6 Mpbs.
- Lucent Wavelan IEEE 802.11b cards, which allow for (nominal) throughputs of

Form factor	PC card type-II extended
Bit Error Rate (BER)	better than 10^{-5}
Nominal output power	15 dBm
R-F frequency band	2.4 GHz (2400-2500 MHz)
Number of selectable sub-channels	North America: 11
	Europe: 13
	France: 4
	Japan: 1
	Other countries: 11 or 13
Data Rate	High: 11 Mbps
	Medium: 5.5 Mbps
	Standard: 2 Mbps
	Low: 1 Mbps

Table 5.1: Characteristics of Wavelan cards.

11 Mpbs. They are not compatible with the older IEEE 802.11 models, so they intercommunicate at rates that are common to both, that is, 1 or 2 Mbps. Table 5.1 shows some characteristics of these cards, as provided by the manufacturer.

- Lucent Wavepoint access points, which can use either the old or new card models. These access points are enhanced by an external antenna that increase the carrier's power by 17%, theoretically.

- A Pentium III 450 MHz desktop computer, acting as a static host.

- Two portable computers, AMD K6 350 MHz, acting as mobile hosts.

5.1.2 Performance testing

Although there are several theoretical studies, whitepapers, marketing materials, and other documents that describe the characteristics of wireless networks such as the one we have used for our test platform, we decided to carry out a set of experiments to evaluate the performance of our network under the specific *real* conditions of our environment. As we imagined, the results of these experiments diverge from those found in the literature.

Before describing the first set of experiments we carried out to evaluate the performance of our wireless network, some precisions about the configuration and environment are in order: only one access point (AP) is used, which is located inside an office nearly in the middle of the building. For practical reasons, the AP is placed close to the Ethernet connectors, which are right below the windows; this fact introduces an above-below asymmetry (in the sense of the width of the building), as can be verified in either Figure 5.1 or 5.2. We used a traffic generator tool called `netperf` for sending traffic at the maximum available rate.

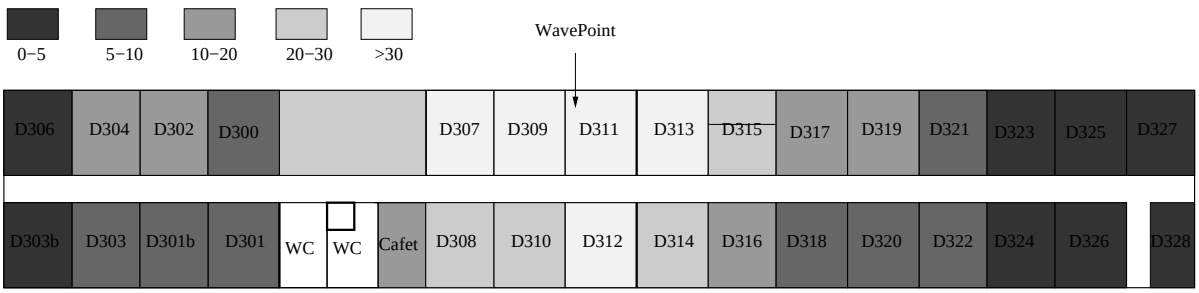

Figure 5.1: Quality of signal as a function of location.

The first experiments have the goal of determining the transmission performance, as received by a static station S_1. Then, the setup is as follows: there is a mobile host M_1 sending traffic to S_1, passing through the AP. This way, all throughput and delay measures are dependent on the quality of the signal that the AP receives. The transmission between S_1 and the AP is done through wired Ethernet and can be considered, for all practical purposes, constant.

Our first observation is the dramatic variation of the quality of the signal that can take place even in the same office: at one point of the office the quality can be very good, but it can degrade to unacceptable levels less than two meters away. We attribute this to interferences that occur between the AP and the mobile station when the signal is reflected by an obstacle. Noise was also measured and we observed low levels that have insignificant variations over time and space; for this reason we will only take into account the signal to noise (S/N) ratio. Transmission delay also shows insignificant variations in the temporal and spatial dimensions. We measured a base delay of 2 milliseconds, with small random variations.

Figure 5.1 shows different signal qualities observed according to the position of the mobile station. A strange situation can be immediately seen to the left: signal quality is significantly better in offices D302 and D304 than in D300 (it should be noted that the experiment was repeated several times to verify these results). The right-left asymmetry due to the different types of walls is very notorious, as is the above-below asymmetry due to the position of the AP.

We also observed a maximum effective throughput of 5 Mbps. It is relevant to point out that our wireless network is connected to an Ethernet network (10 Mbps) that has a maximum effective throughput of 8 Mbps. Figure 5.2 shows the throughput as a function of the position in the building (with respect to the AP). It is obvious that there is a correlation with the signal quality. Below 1 Mbps the network becomes unusable. We also made several tests to compare the throughput generated by TCP and UDP traffic. The difference was always practically the same and corresponds to the difference in the overhead of the protocols.

The second series of experiments aims for determining the usable bandwidth when different traffic sources share the medium. First, mobile host M_1 uses `netperf` to send

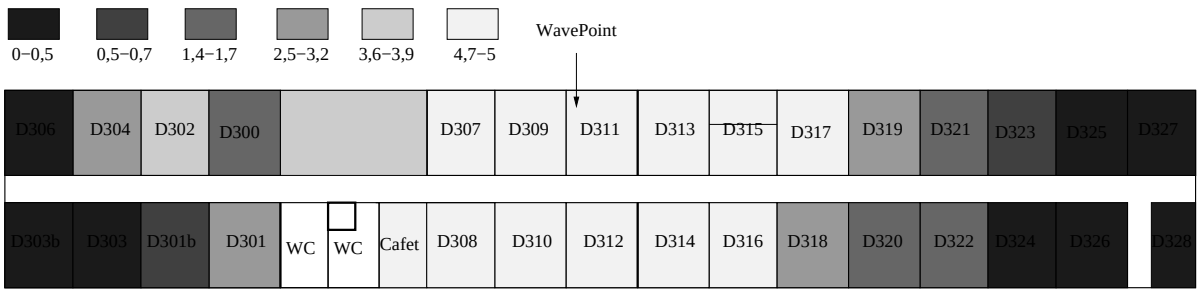

Figure 5.2: Throughput as a function of location.

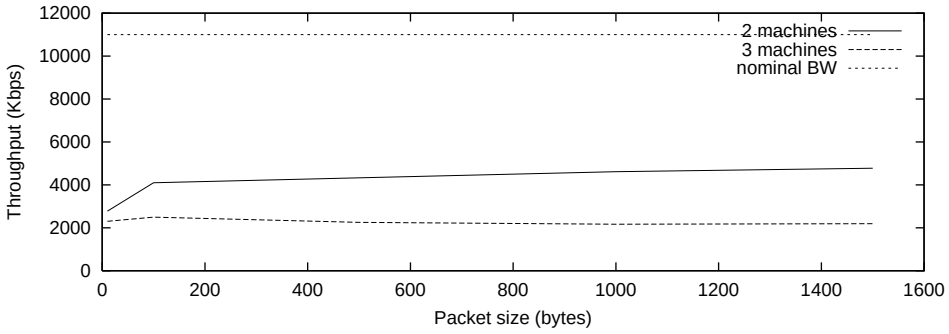

Figure 5.3: Useful bandwidth in a 802.11b WLAN.

TCP traffic to fixed host S_1. As S_1 sends ACKs in response to M_1's traffic, two traffic sources are then sharing the medium. A third source, (mobile) host M_2, is then introduced sending traffic to S_1. Figure 5.3 the effective throughput for these experiments.

In another set of experiments, we wanted to determine what happens when two stations communicate using different rates. We found that as repeated frame drops are detected — due to factors such as distance from the AP, interferences, attenuation, etc.— the Wavelan equipment degrades the bit rate (*e.g.*, the 802.11b cards degrade from 11 Mbps down to 5.5, 2, or 1 Mbps). In addition to that, since the probability of access to the medium is equal, hosts capable of sending at a high bit rate are negatively affected by hosts that send at a low rate. Table 5.2 shows the measured performance; throughput is measured at the TCP layer. When both hosts transmit at the high rate of 11 Mbps, they are able to do so with a throughput slightly superior to 5 Mbps[1]. However, when a host transmits at the high rate and the other at the low rate, the high rate host is obliged to send at a throughput that the low rate host can sustain. Even if the high rate host has the capacity to transmit with a higher throughput, it has to adapt to allow the principle of fair access to the medium.

[1]The announced nominal bandwidth of 11 Mpbs is reduced in reality due to protocol overhead (MAC, network, and transport), fading, interferences, etc.

host rates	measured throughput
11 Mb/s, 11 Mb/s	5.2 Mb/s
11 Mb/s, 1 Mb/s	0.84 Mb/s

Table 5.2: Hosts of different rates and measured throughput

5.1.3 Impact of the WLAN on QoS

The performance characteristics of the WLAN have a direct impact on the possibility of providing QoS guarantees. It is clear that if the lower layers do not allow an efficient and reliable communication to take place, the higher layers —and ultimately the user— will not be able to receive good service quality. Technologies like ATM were designed to provide QoS guarantees *under the assumption of using a medium with low error rate* [59]. Thus, in order to be realistic, we have to make some assumptions about providing QoS guarantees using a 802.11 WLAN:

1. **No hard QoS guarantees are possible**. All wireless technologies are inherently unreliable. The medium is prone to negative factors such as noise, interferences, attenuations, fading, etc. Our experiments have confirmed that the quality of the signal varies considerably in the temporal and spatial dimensions. With an unreliable medium, it is not possible to give hard QoS guarantees.

2. **The number of hosts using the channel has to be limited**. The WLAN uses a shared medium and all hosts are competing for the available bandwidth. Furthermore, important overhead is introduced by backoff times and waiting periods (*e.g.*, SIFS, DIFS) where stations are not allowed to send traffic into the medium. So, the management of resources, *i.e.* bandwidth, depends directly on the number of hosts and their traffic.

3. **The geographical area of movement has to be limited**. Another negative characteristic of wireless networks is that the signal quality is inversely proportional to the distance between the source and the destination. Low signal quality generates frame drops and, as our experiments have evidenced, not only the bit rate is reduced by the radio equipment, but also high rate hosts are degraded to transmit equally as low rate hosts. So, in order to avoid this degradation, all hosts should move within the zone where they can transmit at the high bit rate.

4. **Traffic sources have to be constrained**. CSMA/CA aims for allowing all hosts fair access to the medium. In order to provide QoS, unfairness has to be introduced, that is, some hosts need higher priority than others. Traffic controls (*e.g.*, traffic shapers) have to be introduced to allow different resource (bandwidth) allocations to different traffic sources (hosts).

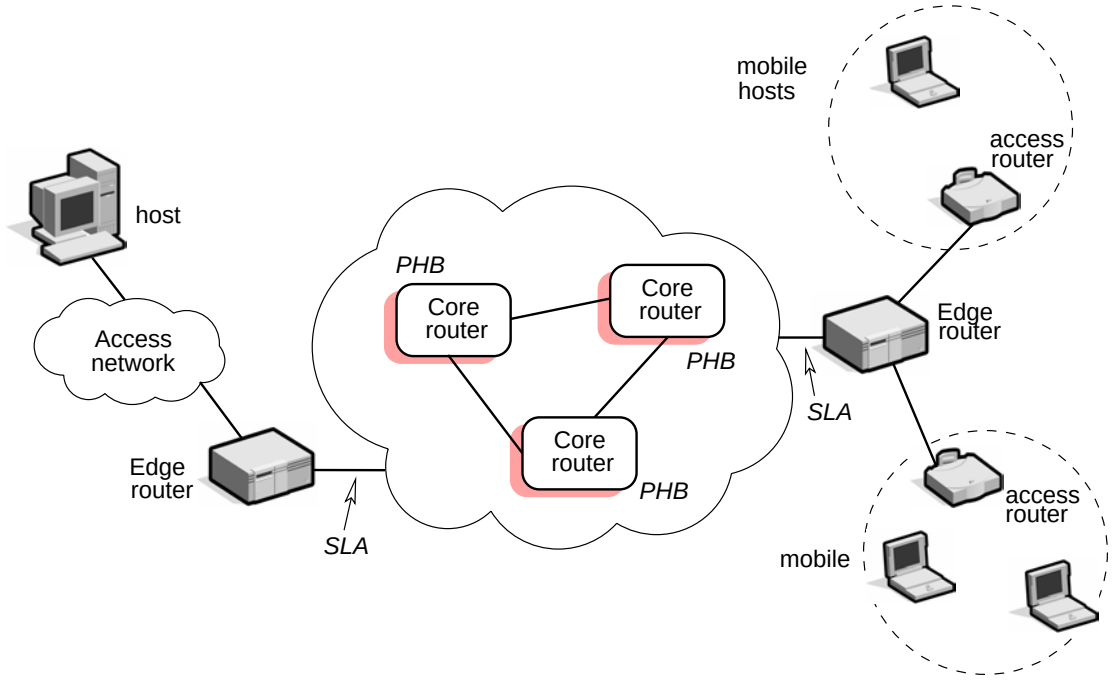

Figure 5.4: A differentiated services architecture.

5.2 QoS Management Solutions

As seen in Chapter 3, IntServ and DiffServ are the most widely accepted approaches for QoS management in IP-based networks. Serious problems have been detected with IntServ, among them, scalability problems due to the per-flow signaling used by RSVP. If these problems hold for wired networks, they are much worse for wireless networks, given their highly dynamic nature. Recent work exploring the use of IntServ solutions in wireless access networks has indicated that QoS signaling generates too much overhead and that the solution does not scales well [93; 6]. Also, IntServ aims for the provision of hard guarantees and, as our experiments have evidenced, these types of guarantees can not be granted in WLANs.

Given the characteristics of wireless networks and our experimental conclusions (*cf.* section 5.1.3), we propose the use of the much simpler DiffServ model. In WLANs, no tight bound on performance measures can be offered and thus the statistical guarantees offered by DiffServ would be more realistic. There is another pragmatic side-benefit with this choice: as DiffServ is increasingly being deployed in the global Internet, its deployment in the edge (access) networks will provide consistent end-to-end QoS behavior, without the need for mapping between QoS classes of different models. The IETF has proposed the DiffServ model for over-provisioned networks, but we propose applying it in WLANs following certain conditions presented earlier. One of this conditions implies the use of traffic controls to allow bandwidth allocations for different traffic sources.

5.2.1 Our differentiated services architecture

As discussed in chapter 3, the DiffServ model is based on flow aggregation and per-hop behaviors applied to a set of classes. The IETF has defined two types of Per-Hop Behaviors (PHB) called EF (Expedited Forwarding) [70] and AF (Assured Forwarding) [62].

Within the framework of the AIRS project [1], we have adopted a differentiated services architecture that defines two PHBs loosely resembling EF and AF, and a default PHB that provides a BE service. Thus, using these PHBs, we define three classes of services (guaranteed, assured and default). The idea of using three classes of services has also been proposed in [105]. Our proposed service classes are:

- *GS (Guaranteed Service).* A service using the EF PHB. Defined for flows that need strict QoS guarantees and should be provided with small delay and jitter and a low packet drop rate. Thus, GS packets get the highest priority. GS flows are envelope multiplexed: waiting probability of GS packets is kept low by controlling the number of admitted flows based on their peak rate and by providing enough resources (link capacity).
- *AS (Assured Service).* A service using the AF PHB. Meant for elastic flows that do not have strong constraints in terms of packets delay and jitter, but need a minimum guaranteed average bandwidth. The bandwidth for this service is divided in two parts:
  - a fixed part that corresponds to the minimum assured bandwidth. In case of network congestion the packets in this part are marked as not eligible for discarding.
  - an elastic part with opportunistic packets. There are not any guarantees for these packets that are forwarded on a best-effort basis. In case of congestion, these are the first packets to be dropped. The opportunistic bandwidth varies according to resources availability, showing an elastic behavior.
- *BE (Best Effort).* A service that does not offer any kind of guarantees.

The GS and AS classes allow a qualitative differentiation between flows according to their QoS requirements (*e.g.*, transit delay, minimum bandwidth). It should be noted though, that each one of these service classes can originate a great variety of services which can be defined according to a specific Traffic Conditioning Agreement (TCA). Figure 5.5 shows an example of bandwidth sharing between GS, AS and BE flows in a certain time-frame.

5.2.2 QoS mechanisms for core and edge behaviors

As previously discussed, core and edge routers have well defined functionalities within the global Diffserv architecture (*cf.* Figure 5.4).

Edge Router

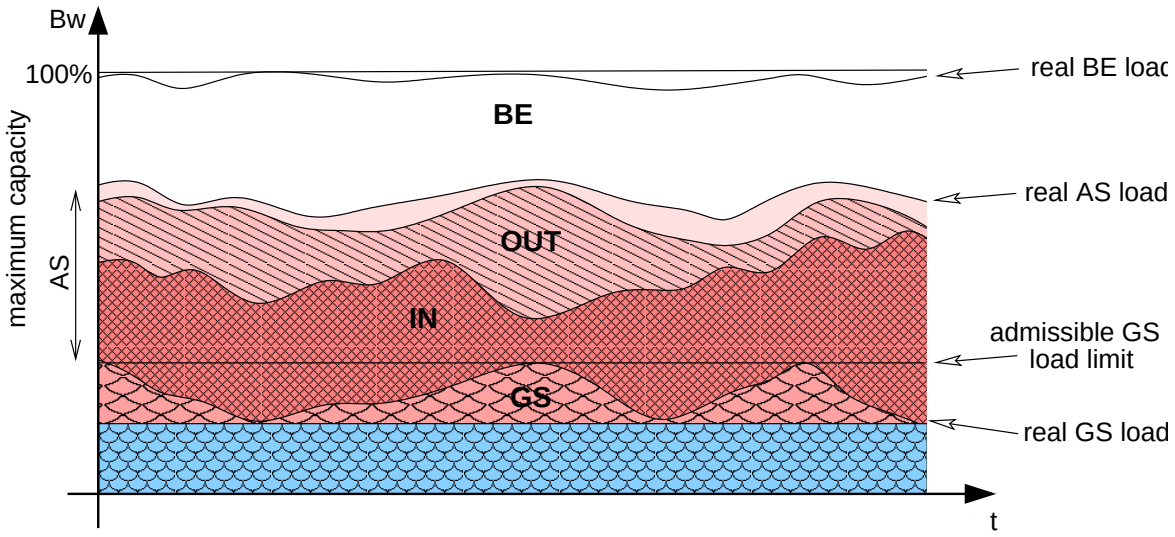

Figure 5.5: Bandwidth sharing between GS, AS and BE.

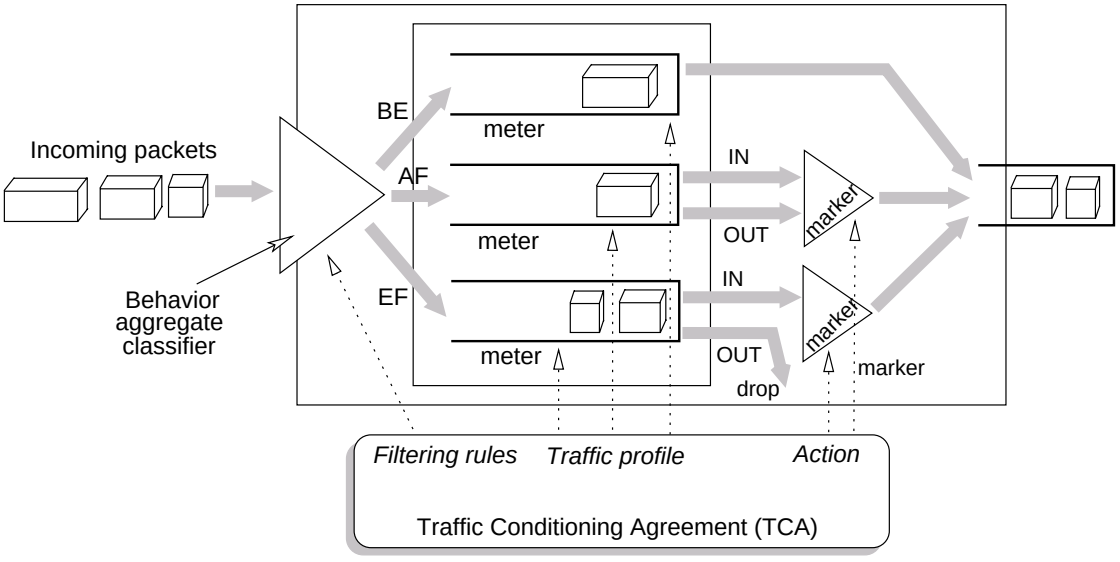

Figure 5.6: Architecture of an edge router.

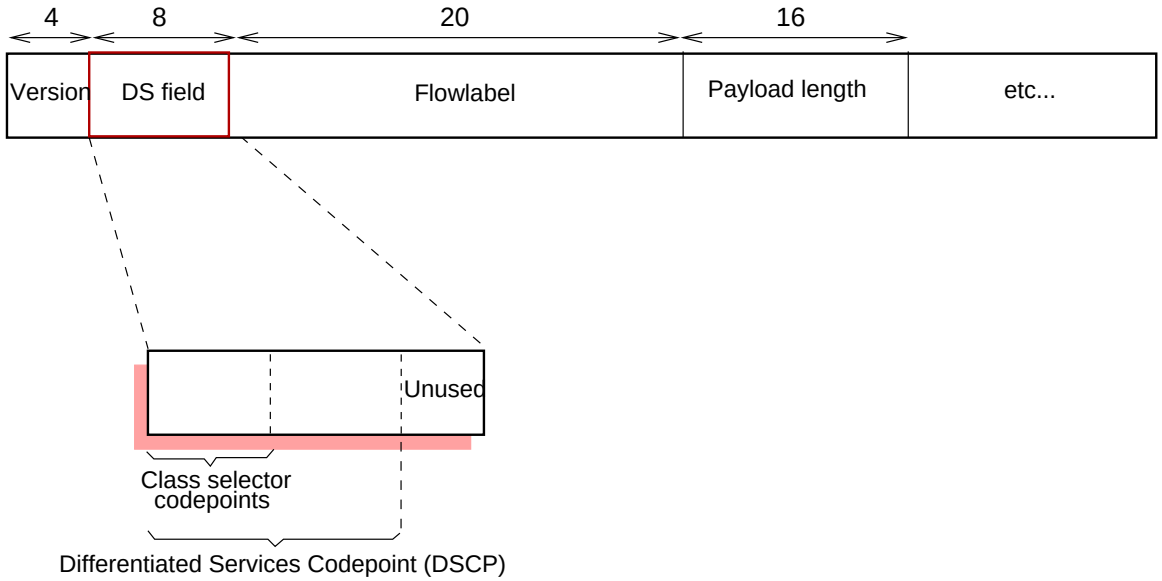

Figure 5.7: Format of the IPv6 DS header field.

The edge router is in charge of packet classification and marking, as well as of conditioning the incoming traffic. Rules and parameters for traffic conditioning are defined in a TCA, which contains filtering rules for identifying the service's user, traffic profile, actions for exceeding traffic, etc., as shown in figure 5.6.

Following the AIRS project model [1] we have adopted the token bucket control (*cf.* section 3.1) for verifying profile compliance. For GS traffic the profile consists of only a peak rate. Verification of TCA compliance consists then of a meter-spacer, which is a queue of finite size. Accepted packets are marked as EF and rejected ones are dropped. The queue allows packet bursts by delaying packets so they can respect the peak rate. Thus, it is important that the size of the queue be small enough so that waiting delays stay acceptable. For AS flows, the profile consists of a mean rate and burst tolerance –two parameters of a leaky bucket. TCA compliance is verified by a leaky bucket; packets within the profile are marked as *prioritary AF* and out of profile packets are marked as *non-prioritary*. In the case of BE flows no control at all is enforced.

As mentioned above, non-rejected packets are marked; this marking is performed according to RFC 2474 [105], which uses the DS field in IPv6 headers, shown in figure 5.7. Only 6 bits, named DSCP (Differentiated Services Code Point), within the DS field are used for marking, and for selecting the PHB in core routers. In order to keep compatibility with IP precedence [5; 8], the DSCP has a relative priority coded in the leftmost 3 bits, which act as a *Class Selector Codepoint*, indicating forwarding priorities between PHBs.

Core Router

Figure 5.8 shows the architecture of core routers. When packets arrive at the core router they are first classified according to their class of service, as indicated in their IPv6 DSCP header field; then they are inserted in the corresponding queue, which have different

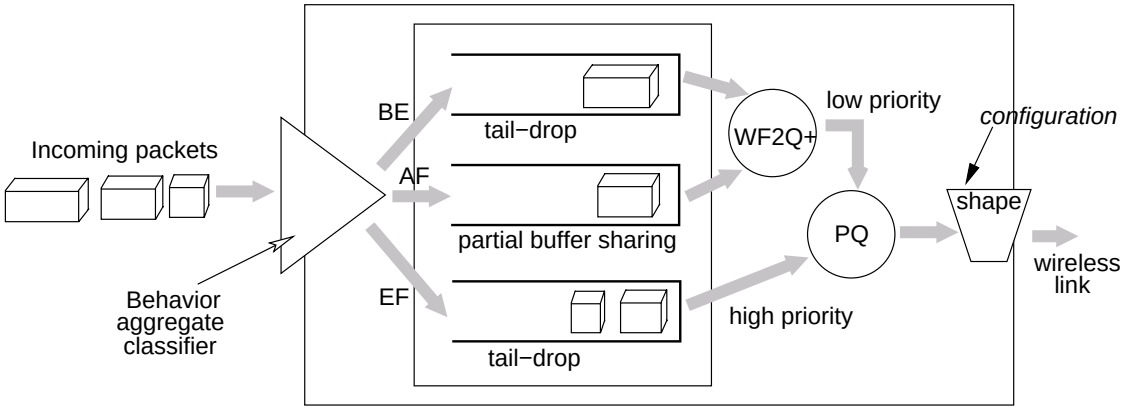

Figure 5.8: Core router architecture.

behaviors. EF and BE queues use a tail-drop policy: a packet is dropped when the queue is full. Packets in AF flows are subject to the PBS (Partial Buffer Sharing) policy: only conformant packets are accepted when the queue is greater than a given threshold. This way, all AF packets may benefit from available resources, however in case of congestion only conformant packets will be allowed in the network. AF and BE classes are scheduled using the WF2Q+ (Worst-case Fair Weighted Fair Queueing) algorithm [10; 11], which is a variant of WFQ (Weighted Fair Queueing). Then, high-priority packets (EF) and low-priority ones –those coming from AF and BE– pass through priority queueing.

5.3 Mobility Management Solutions

As stated in Chapter 4, mobility in locally restricted areas (micro-mobility) should be handled differently than global mobility. The effect of handovers, in terms of latency and packet loss, should be limited in order to improve overall performance. We are also assuming the use of a local protocol that will operate in a restricted domain, thus reducing the amount of signaling needed and having a positive effect on performance. This protocol will be designed keeping in mind the smooth integration with QoS support.

Another essential assumption we make is that, during local mobility of hosts performing handoff from cell to cell, such *hosts should keep their routable IP address*. This contributes to another performance enhancement, since there is no overhead due to new address assignment or registration with foreign agents. Equally important is the fact that there is not any negative impact on the higher layers, notably the application one, present on the mobile device or in the network. Keeping the same IP address is made possible by the careful preparation of a new route that will reflect the new location of the host.

A micro-mobility solution, such as the one we propose, which localizes processing and signaling, constitutes a light-weight approach customized to the exact requirements of local mobility, but does not exclude the use of other global mobility mechanisms (*e.g.*, Mobile IP [118]). Moreover, our propositions are IP version neutral and layer 2 technology

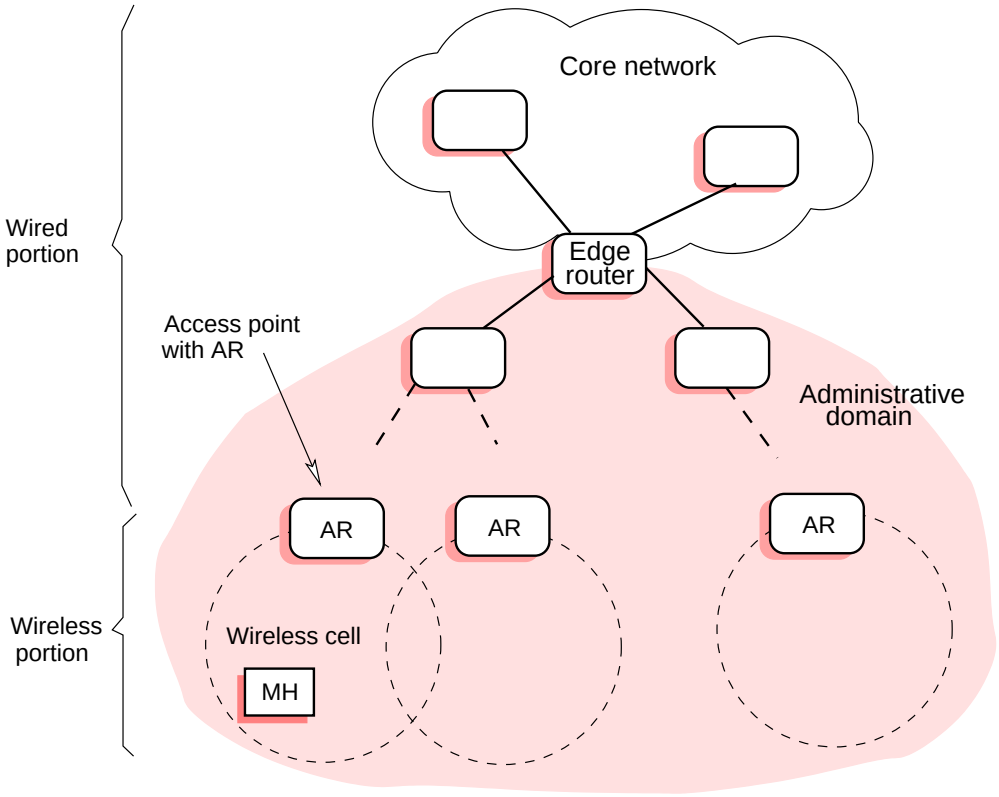

Figure 5.9: Elements of our mobility architecture.

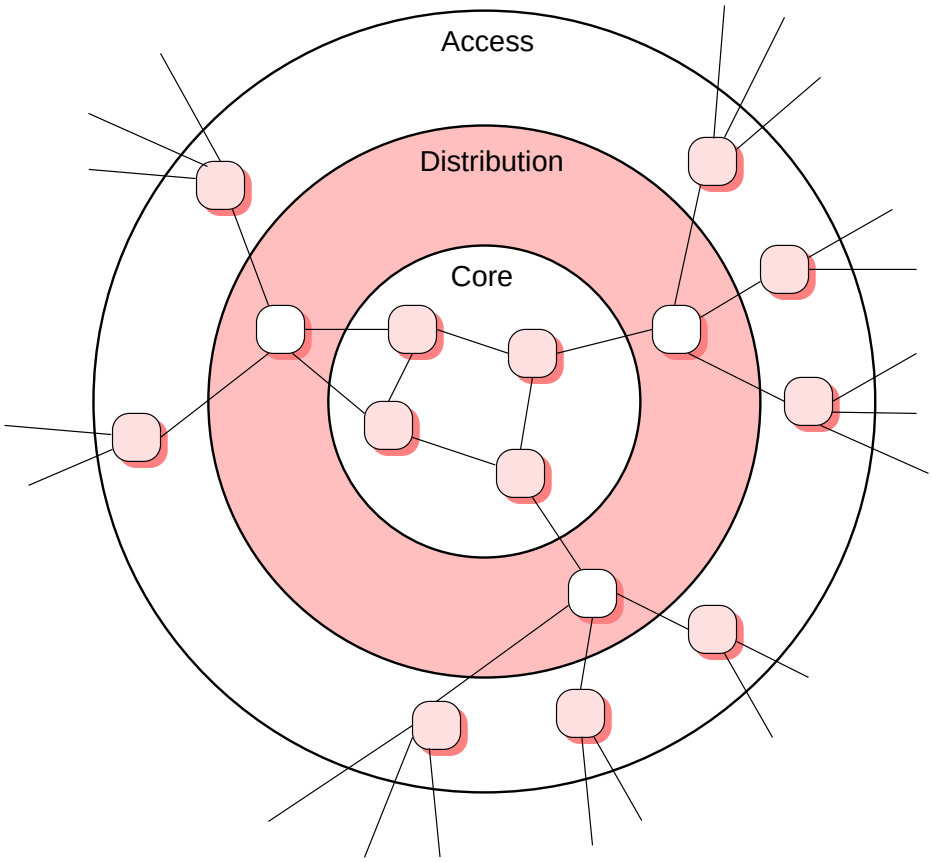

Figure 5.10: Structured hierarchy: core, distribution, and access.

independent, although implementation details may vary. This constitutes an added value for deployment across heterogeneous networks.

Before continuing with the description of our protocol, we will give definitions and detail characteristics concerning the elements of our mobility architecture (depicted in Figure 5.9), most of which are standard definitions found in the literature:

- *Structured hierarchy.* A common practice for networks is to maintain strict levels of hierarchy, commonly referred to as *core*, *distribution*, and *access* levels (Figure 5.10). This limits the degree of meshing among nodes. The core portion of the hierarchy is generally considered the central portion, or backbone, of the network. The access level represents the outermost portion of the hierarchy, and the distribution level represents the aggregation points and transit portions between the core and access levels of the hierarchy [48].

- *Administrative domain.* A collection of networks under the same administrative control and grouped together for administrative purposes [115].

- *Edge router.* A router on the edge or periphery of an administrative boundary or domain.

- *Intermediary routers.* Those found in the way between access and edge routers.

- *Access router.* A router residing in an access network that offers connectivity to mobile hosts. The router may include intelligence that goes beyond simple forwarding service offered by ordinary IP routers. Note that, for our conceptual architecture, access routers are co-located with access points[2], although for implementation purposes both could be physically separated.

- *Cross-over router.* The router closest to the mobile host that is at the intersection of two paths: one between the edge router and the old base station, and the second between the old base station and the new base station. Under particular situations the cross-over router may be located at the edge router, but this is not a general case.

- *Mobile host.* Also known as mobile station. An IP node capable of changing its point of attachment to the network.

- *Wireless cell.* Also known as radio cell. An area associated with each access router, covered by the radio channel.

5.3.1 Mobility protocol

Simplicity was one of the design requirements for our mobility protocol. It basically consists of an initial handoff request from the mobile host to the new access router, that in turn will trigger a series of handoff acknowledgments going up to the cross-over router and down to the old access router. Each time an acknowledgment is issued, the involved router changes its forwarding table in order to forward packets destined to the mobile through the interface on which the acknowledgment arrived.

In order to better understand how our protocol works, we will give an example of its functioning during a handoff. Figure 5.11 depicts a situation where a mobile host M_0 tries to handoff from the old access router AR_1 to the new one AR_n. There can also be an arbitrary number of intermediary routers between an edge router and the cross-over router (R_c). Each router can have several interfaces, and we are just showing interfaces A and B for the access routers, and interfaces A, B, and C for the cross-over and intermediary routers. Next to each router there are indications of what interface is used to forward packets to M_0, before and after a protocol message is received. So, during a handoff the following actions take place:

- *Handoff initiation.* A certain event makes M_0 decide to move to another cell; such event could be reaching a lower threshold of the signal to noise ratio, or taking into account some other parameters such as the load or number of hosts in the current and adjacent cell. A handoff request message (`HO_REQ`) is then sent, and traverses the wired infrastructure (passing through AR_1) to finally reach the selected access router (AR_n).

[2]Also known as base station, is a layer 2 device offering the wireless connection to the mobile hosts.

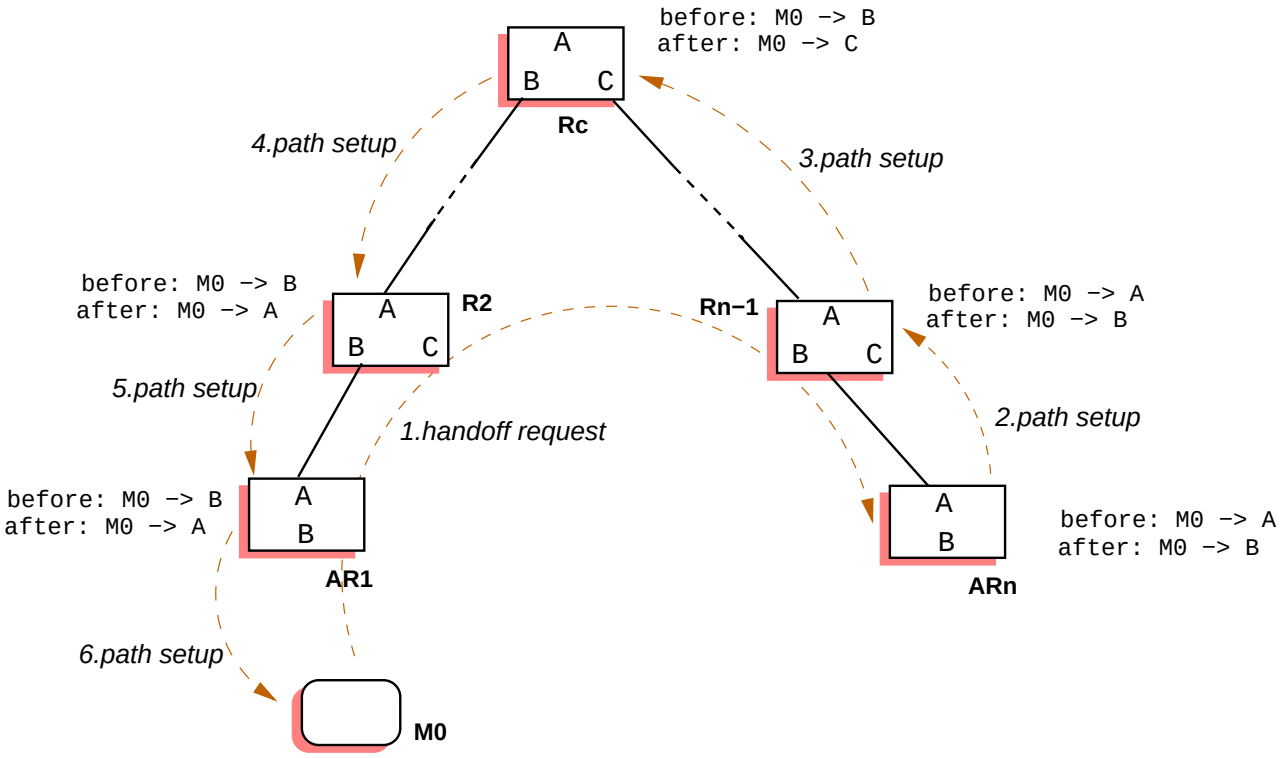

Figure 5.11: Mobility protocol messages during a handoff.

- *New path setup propagation.* Once AR_n receives and accepts the handoff request, path setup messages (`HO_ACK`) are propagated hop by hop, starting from R_{n-1}, eventually reaching R_c, and then going down to AR_1. These messages take into account the fact that M_0's next hop router will be AR_n, and no longer AR_1, so forwarding entries in all involved routers should be modified to reflect this fact. For instance, when the message is received by R_{n-1} from AR_n, a new route in R_{n-1}'s table is inserted to indicate that packets for M_0 will be forwarded to interface B (which allows to reach AR_n). It should be noted that after this change is made, any correspondents "behind" interface C will be able to reach M_0 via its new route, which passes through AR_n. An analogous observation applies to all other routers each time they perform new path setup.

- *Handoff completion.* New path setup messages will eventually reach AR_1, that will in turn send one to M_0. This will mean that the new path has been completely setup and that M_0 should now change its routing table, by specifying the target access router AR_n as its default router, and should from now on operate exclusively in the radio channel frequency used by its new base station.

Our protocol discussion up to now has described a "normal" or typical operation where, for instance, the mobile host is able to listen to both the old and new base stations during a short period of time (long enough to complete the handoff). We have also described a situation where new path setup is completed without problems, as there were not any

kind of disruptions during the process. Our protocol is, however, capable of handling exceptional conditions, which contributes to its reliability. We will now describe some of these reliability features.

- *Rollback.* Once AR_n receives a `HO_REQ` message, it starts a handoff procedure by sending a `HO_ACK` message that will be propagated through all involved routers, change their tables, until eventually reaching M_0 to complete the handoff. However, if along the path a router is not able to correctly set its table, a `HO_NACK` message will go back through the same path (towards AR_n) advising routers to undo previous changes; the `HO_NACK` will also be sent towards AR_1 to eventually reach M_0 and indicate that handoff can not take place[3].

- *Dual-casting.* Our example above assumed that the radio equipment used by M_0 is such that it will have the possibility of communicating with both the new and old base stations for a short period of time. This is actually the case with some equipment currently available in the market, such as 802.11-compatible devices that allow this mode of operation[4]. Nonetheless, our protocol is layer 2 independent and using this type of operation is not a rigid assumption. We assume the use of an optional dual-cast mode where each time a router changes its forwarding table, it will not only send traffic to M_0 through the new interface, but will keep also sending —for a short period of time— this traffic through the old interface. A simple rule is observed, though, in order to avoid loops: packets should not be re-sent through the interface they arrived from. An actual implementation of the protocol could have the ability to trigger on/off the dual-casting mode, depending on the characteristics of the available layer 2 equipment.

- *Mobility bootstrap.* When the mobile host is first turned on, it has to "associate" itself with an access router. This is handled by an extension to our mobility protocol, called reverse paging, that will be discussed in section 5.4.1.

5.3.2 Discussion and highlights

During the development of our mobility management scheme, we compared several other existing ones and considered the use of Hawaii's UNF and MNF schemes. However, they did not incorporate some of our design requirements, such as the smooth integration with QoS support. Contrary to Hawaii, where the new access router is contacted directly to start new path setup, we proceed to first contact the old access router. This way we foresee that QoS mechanisms could have the possibility of first checking whether enough resources will be available in the new cell, before proceeding to handoff.

The order of route updates prevents transient routing loops or creation of multiple traffic streams during handoff, similarly to Hawaii's UNF and MNF schemes. As the route updates are done before the mobile host changes the transmission channel, it receives all packets along the old route.

[3]this assumes that routing nodes are "up" and can send these messages, however, the case of the nodes being "down" is still an open issue

[4]In this period of time the radio equipment enters what is called "promiscuous mode".

Moreover, the scheme is optimized so that the traffic is delivered as soon as possible to the new location: after the new path setup starts taking place, beginning at the new access router, some traffic can start being delivered through the new path. However, if the layer 2 equipment does not allow the mobile to listen to both the new and old access routers, packets going along the new path may be sent by the new access router before the mobile changes its radio channel and is able to receive them; loss may then occur during a short period between the instant of the path update and the beginning of the communication in the target cell. This could of course be avoided using the dual-cast mode.

5.4 Towards an Integrated QoS and Mobility Management Architecture

We have already described our proposals for managing mobility and QoS in local wireless networks. We argued that both are related and as a result our mobility management proposal has been designed to smoothly integrate with QoS support. Going a step further, we now propose to integrate our (out of band) mobility management protocol with QoS support. This is done by incorporating a lightweight in-band protocol and what we have called "reverse paging". The proposed integrated architecture will be described and claims as to the expected benefits will be given.

Before going on with the description of the integrated architecture, we should mention three cases of micro-mobility we are interested in:

1. *Active.* The mobile host moves while currently sending or receiving traffic from other nodes. In this case, it is very important to keep latency and packet loss to a minimum while performing handoff.

2. *Idle.* Although fully able to send and receive IP traffic, the mobile does not currently have any inbound or outbound flows. Thus, latency and packet loss during handoff is less of a critical issue.

3. *Dormant or off.* In both cases, latency and packet loss is not a critical issue. While off, there is absolutely no activity going on. While in dormant mode, the mobile restricts its ability to receive normal traffic by reducing monitoring of radio channels. There are implementations of dormant mode in which the mobile alternates between periods of not listening for any radio traffic and listening for traffic.

5.4.1 Reverse paging

Paging is a mechanism widely used in mobile telephony [130], where base stations periodically send signals in order to determine the location of mobile phones. Similar mechanisms have also been recently adopted in mobile data networks; for the specific case of IP-based

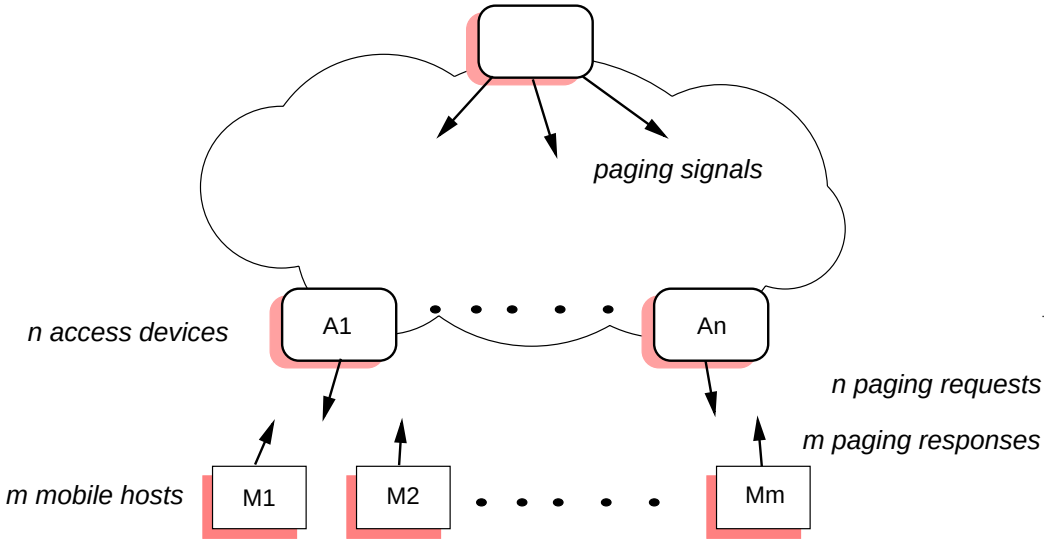

Figure 5.12: Host location signaling in paging.

networks there is a mechanism called IP paging being developed by the IETF's Seamoby working group [77; 78].

Paging implies that the network periodically performs signaling in order to locate mobile hosts. As one of our design requirements is the utilization of a lightweight approach that minimizes signaling, we find that paging has too much signaling overhead and is not suitable for our purposes. We propose the use of another scheme that we have named reverse paging. Our proposition is that instead of the network signaling to locate the hosts, the mobile hosts should be the ones sending signals to inform the network of their location. Hosts will only send signals when a relevant event (such as change of location) occurs. It is also natural that hosts send their signals when they are able to do so (*e.g.*, after waking up from dormant mode). In normal paging, signals are sent even if the hosts are dormant or off. For these reasons, signaling is reduced with reverse paging.

Signaling is also reduced in typical situations such as the case when mobile hosts are trying to be located. Under normal paging (figure 5.12), a node in the network (*e.g.*, a domain root router) sends periodic signals down to the access routers (or other access devices). Assuming that these signals are made of just one packet, there will be as many signaling packets in the wireless part of the network as there are access devices (n in this example). Then, if there are m mobile hosts capable of listening at that time, there would be m responses indicating their location. Thus, there would be $m + n$ packets traversing the wireless infrastructure to perform host location. In reverse paging (figure **??**), the location is not polled by the network, so there would only be m messages sent by the mobile hosts indicating their location. Note that this is a worst-case scenario when all m hosts wake up simultaneously, or for some other reason (*e.g.*, a pre-programmed signaling interval), decide to sent their location advertisement at the same time. Typical situations involve less than m messages traversing the network simultaneously.

Reverse paging constitutes an extension to our mobility management protocol. For

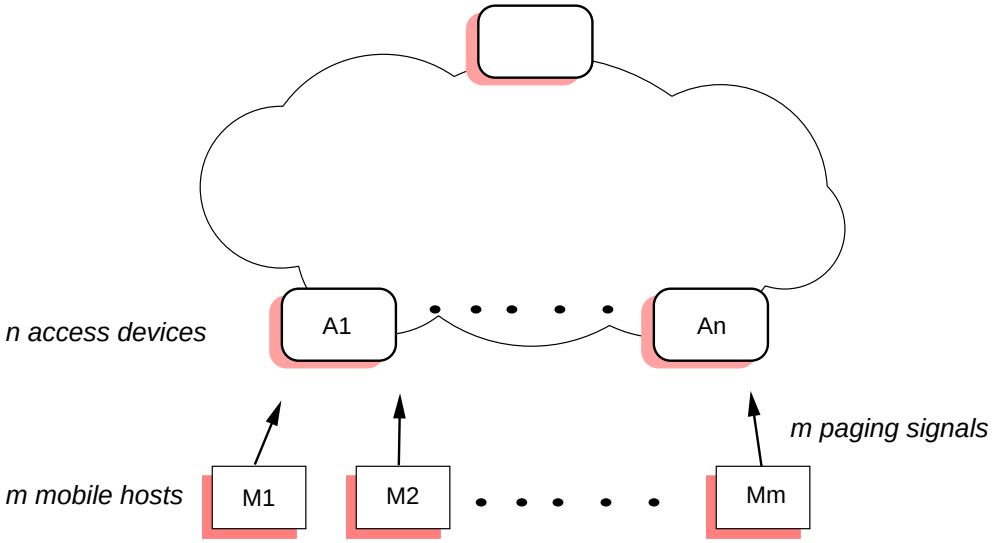

Figure 5.13: Host location signaling in reverse paging.

example, when a mobile host moves while capable of sending/receiving traffic (*i.e*, either in active or idle mode), then the location advertisement message will in fact constitute the event that will trigger our handoff procedure (handoff request equals location advertisement). Reverse paging will also handle the case of micro-mobility in idle mode or while turned off. Indeed, when the host wakes up, it will contact an access point within communication range to send a location advertisement. This way, a normal handoff procedure will be initiated.

5.4.2 In-band signaling

The approach of using in-band signaling on mobile data networks has been explored in the ATM [99] and IP [90] contexts. The basic idea with in-band signaling is to use normal traffic to insert control data inside packets (normally within headers). With this in mind, we propose to use not only normal data packets, but also control packets used for mobility management (including reverse paging) to insert data in IP headers that will allow us to control our QoS mechanisms. We are effectively creating this way an in-band protocol that will be coupled to our (out-of-band) mobility protocols, to create an integrated architecture for QoS and mobility management. Protocol commands are encoded in packet headers.

It is worth highlighting that our proposed in-band protocol has a local scope restricted to the micro-mobility domain, *i.e.*, nodes outside this domain have no obligation of interpreting the protocol commands, and may even re-write them. Since we use a DiffServ architecture where there is no flow classification (like in IntServ) but rather behavior aggregation, the IPv6 flowlabel field is normally unused, and we propose to use such field for in-band protocol commands[5]. As seen in Figure 5.14, we propose a simple format

[5]In fact, the proposal is also valid for IPv4, where the Options field could be used.

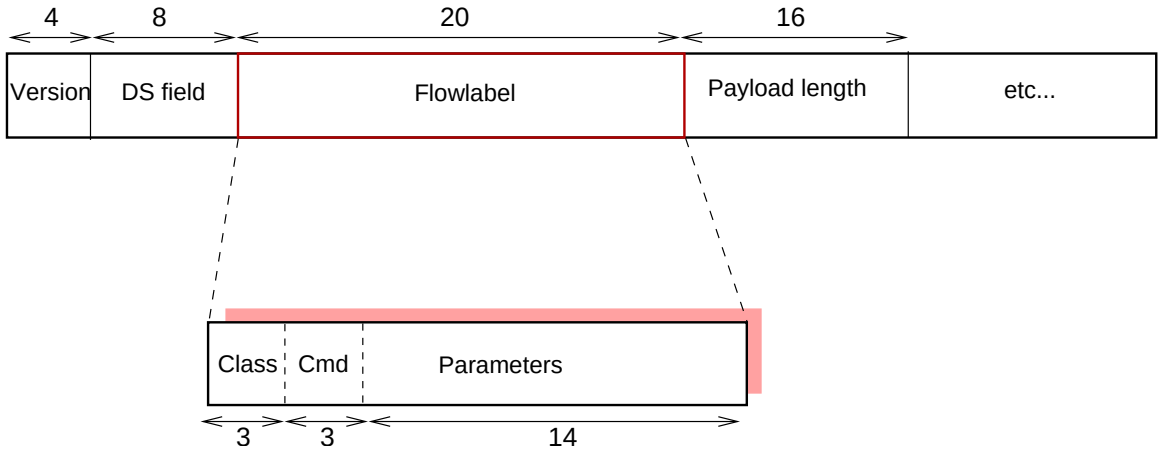

Figure 5.14: Use of the IPv6 flowlabel field for in-band signaling.

Command	Parameters
`set_bw`	`class` (3 bits), `bandwidth` (11 bits)
`beacon`	`type` (3 bits), `status` (11 bits)

Table 5.3: Some uses of the command and parameters sub-fields for in-band protocol commands

for the semantics of this 20-bits field: 3 bits for indicating the QoS service class of the packet, 3 bits for coding the protocol command, and the remaining 14 bits for command parameters.

Protocol command's parameters may have different formats and semantics depending on the type of a command. In Table 5.3 we can see that the set bandwidth (`set_bw`) command takes two parameters: a 3-bit `class` and a 11-bit `bandwidth` parameters used to indicate the class for which a certain bandwidth should be set up. An actual implementation could use the `bandwidth` field to represent bandwidth "slices" of, say, 64 Kbps. Another type of command, of an informational nature (called `beacon`), will either request or give information on the status of QoS parameters. This way, 3 bits are used to encode the type of beacon (request, informational) being sent and the remaining 11 bits will be used for optional beacon parameters (some beacons may not have any). Some command examples (Figure 5.15) may be useful: to indicate that 128 Kbps should be set for the GS class, the command sub-field is set to `001` (`set_bw`), the `class` parameter is set to `010` (GS class), and a binary 2 is coded into the `bandwidth` parameter to indicate that two "slices" of 64 Kbps should be allocated. A `beacon` command can be encoded using `010` and having as a unique parameter a bandwidth availability status request (`bw_status`) represented by `011`.

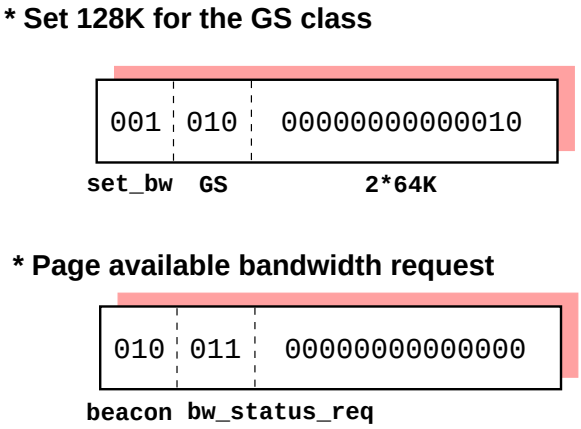

Figure 5.15: Examples of in-band protocol commands.

5.4.3 QoS and mobility management integration

We have designed our mobility and QoS management solutions with their smooth integration in mind. The out-of-band mobility protocol (including reverse paging) is coupled with in-band QoS management protocol to obtain an integrated architecture. This way, mobility and reverse paging protocol packets carry in-band QoS protocol commands that are encoded inside header fields. Moreover, there is a relation between in-band commands and out-of-band commands. For instance, packets sent from time to time to maintain soft-state will carry QoS status request or inform commands; handoff request (`HO_REQ`) may carry QoS requests, and handoff acknowledgments (`HO_ACK`) may carry QoS acknowledgments. This implies the presence of several entities (shown in Figure 5.16) collaborating to do monitoring and determine when to issue commands, and modifying QoS behaviors and routing tables. Among them, we can mention the following:

- *Configuration manager.* A coordinating entity that configures the behavior and parameters fo the packet classifier (*e.g.*, classes, destination ports, etc.), the QoS monitor (*e.g.*, minimum signal quality threshold, highest cell load, etc.), and the mobility monitor (*e.g.*, soft-state refresh interval, etc.)

- *Packet classifier.* It marks packets to indicate that they belong to one of the predefined QoS classes. The policies that indicate how to classify the packets are usually set by the configuration manager. Some QoS-aware applications (Q-apps) can mark their packets to classify them, but "normal" applications can not, so it is up to the packet classifier to do it.

- *QoS monitor.* It constantly monitors the status of resources affecting QoS (such as bandwidth). When significant variations are detected, it informs the mobility monitor to help to decide if a handoff should be performed or if beaconing packets should be sent. The QoS monitor injects in-band signaling information into outgoing IP packets to dynamically (re)configure QoS mechanisms (present in edge and core routers). It is also in charge of reading this in-band information from incoming packets and to act accordingly.

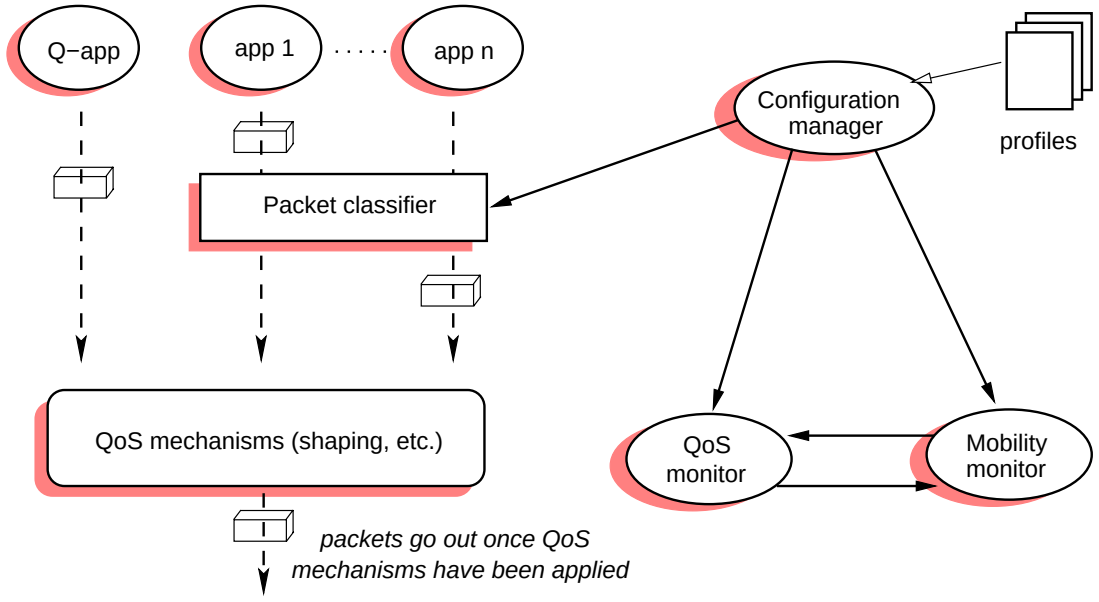

Figure 5.16: Functional architecture for integration of QoS and mobility management.

- *Mobility monitor.* It emits reverse paging commands and starts handoff procedures; it is capable of changing routing tables for performing new path setup. The decision to initiate a handoff is triggered by the information it receives from the QoS monitor regarding the current status of resources (*e.g.*, signal strength, cell load, etc.).

The entities described above are present in all routers (including edge and access routers) within an administrative domain. This way, micro-mobility management and intra-domain QoS management can be carried out by the coordination of the different entities. All protocol commands (in-band and out-of-band) are valid within the administrative domain, but nodes outside the domain may ignore or discard them.

Our architecture manages QoS in a hierarchical way as we distinguish two time scales and two levels of management: intra-cell management and inter-cell management.

- *Intra-cell management.* The management that takes place within one cell. The current status of the cell can change rapidly as applications are either start or stop sending traffics, new hosts join or quit the cell, or hosts move within the cell's coverage. Thus, intra-cell management is very dynamic in nature and it is handled by the corresponding access router.

- *Inter-cell management.* The coordinated management of QoS between cells in an administrative domain sharing the same access router. Global conditions within the whole domain and the relations between the cells change more slowly. This type of management is carried out by the edge router that fixes long-term policies for the access routers.

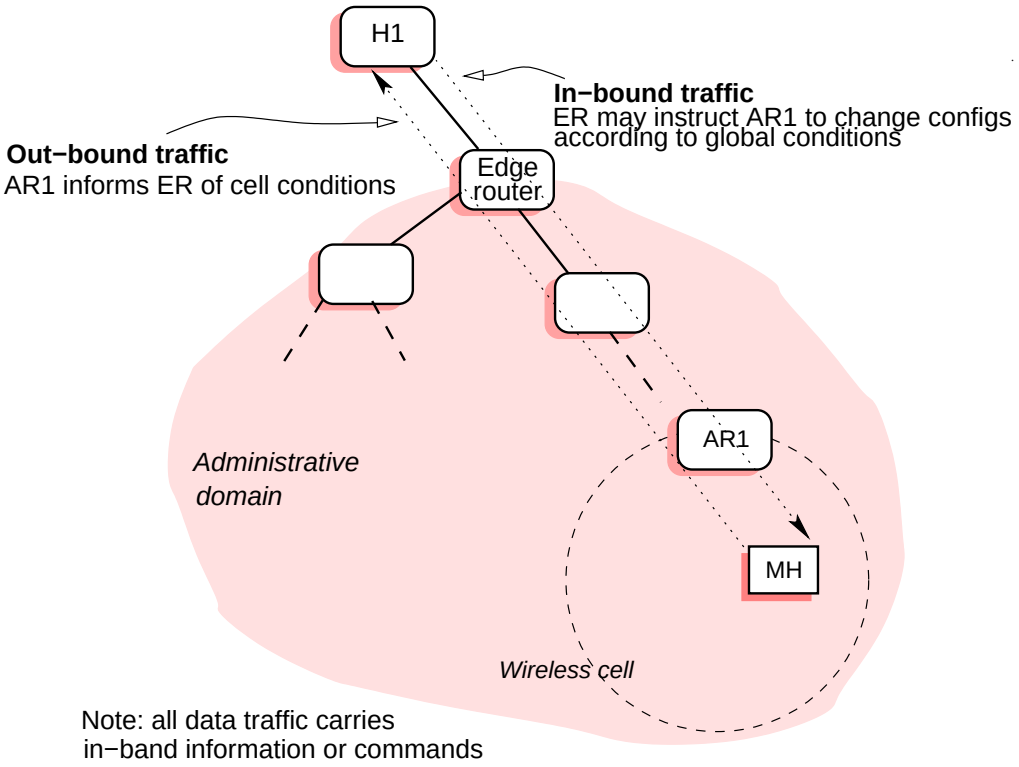

Figure 5.17: Data packets used for QoS management through in-band signaling.

The following scenarios provide descriptions of how this hierarchical management works in typical situations:

Scenario 1

When there is active traffic between a host H_1 outside the administrative domain and a mobile host M_1 in one of the domain's wireless cells, all such traffic passes through the edge router ER. Thus, the data packets are used to carry in-band information regarding the status of resources and commands for changing QoS configurations. For example, packets sent by M_1 to H_1 pass through AR_1, the access router for M_1. When they do so, AR_1 injects a `beacon` in-band command into the packets with a `bw_info` parameter, and a second parameter indicating how much bandwidth (in 64 Kbps slices) is currently available in the cell. ER eventually receives these packets, as they go on their way to H_1, and updates its data structures to reflect the current status of the cell managed by AR_1. When H_1 sends packets bound to M_1, ER inserts a `set_bw` command if, based on the resources status it has received and on the global situation within the whole domain, it judges necessary to modify the configured bandwidth settings for a certain class. This scenario is depicted in Figure 5.17.

Scenario 2

Mobile host M_1 decides to handoff to a neighbor cell managed by the access router AR_2. So, M_1 sends a `HO_REQ` message that eventually reaches AR_2. The `HO_REQ` message contains an in-band `beacon` message indicating the bandwidth settings that M_1 has in

its current cell. AR_2's QoS monitor determines that, according to the current utilization of resources, M_1 will not be able to obtain the same bandwidth after the handoff. Thus, the QoS monitor indicates this to the handoff monitor that decides to deny the handoff request sending a `HO_NACK` message that eventually reaches M_1. This `HO_NACK`, however, contains in-band information indicating the available bandwidth at AR_2's cell, so this way M_1 could decide to send a new handoff request with more a more appropriate bandwidth request.

Scenario 3

A mobile host M_2 is turned on within the reach of AR_1's cell. M_2 starts sending reverse paging beacons in order to establish communication with the closest access router. These paging beacons contain in-band requests indicating the bandwidth needs that M_2 has. When AR_1 receives the beacons it decides, based on the availability of resources, to either allow or deny M_2 to be associated to the cell.

Scenario 4

M_1 is communicating with another mobile host M_2 within the same administrative domain. When M_1 starts a new application, the QoS monitor asks the configuration manager to indicate the QoS allocations that should be set for that type of application. The configuration manager consults the profiles and indicates the QoS monitor the proper values to allocate. When traffic starts flowing from M_1 bound to M_2, all packets pass through AR_1, the access router for the cell. Packets contain in-band `beacon` information that inform of the current utilization of resources. As AR_1 notices a change in resource utilization, it adapts its data structures to reflect this change. However, AR_1 manages intra-cell QoS and if the change is unacceptable, AR_1 may decide to deny access to that traffic and inform M_1 that it should either modify local QoS settings (possibly degrading flows) or not allow the application to send new traffic. If the traffic is allowed to pass, AR_1 modify the in-band information to inform of current resource utilization within its cell. When AR_2, the access router for M_2, receives these packets that pass through, it will update its data structures to reflect the current availability of resources in AR_1's cell. This information could be useful in case a mobile host may want to handoff from cell to cell.

5.5 Chapter conclusions

In this chapter we have proposed solutions for mobility and QoS management in WLANs. Based on the characteristics of WLANs —supported by our empirical results—- we argue that QoS guarantees can only be provided under certain assumptions (*i.e.*, no hard guarantees, limited number of hosts, limited movement area, constrained traffic sources). Henceforth, we state that Diffserv is a suitable QoS provisioning model. Mobility management schemes such as Mobile IP are not adequate for mobility in geographically limited areas and we offer a proposal for mobility management based on a micro-mobility approach. Our mobility protocol was designed to achieve a performance level adequate for the provision of QoS, and even the order of the protocol messages takes into account the integration of QoS mechanisms. Furthermore, we propose an integrated architecture

that takes advantage of our solutions for QoS and mobility management. This integrated architecture uses a light-weight in-band signaling protocol and an extension to our basic mobility protocol (that we have called reverse paging).

Implementation and Experiments

Chapter 6

Implementation and Experiments

In theory, there is no difference between theory and practice. But, in practice, there is.

— Jan L.A. van de Snepscheut

When proposing new protocols, it is very usual to test their functionality, performance, and applicability by carrying out a series of simulations. Special protocol simulation software such as NS [47] can be used for these purposes. While this is a very valid and useful approach, we took a more hands-on (also probably more laborious and lengthy) empirical approach. Our rationale was that, while performing simulations, it is very difficult to take into account all the parameters involved in a real platform, which may directly or indirectly affect the performance and execution of the protocol under test. Factors such as CPU speed, OS processing, I/O overhead, normal working load conditions, radio interferences, signal fading, *actual* transmission speeds and other actual equipment's features and values (as opposed to *announced* values given by vendors), etc., are some of the parameters that most of the time are not properly simulated, or sometimes not even taken into account.

Besides wanting to get a feel of the behavior of our protocol in a typical working environment, we also wanted to start building the foundations of a test platform that will allow us to continue developing our research work and validating it by the implementation of prototypes. This does not means that we are neglecting simulations; on the contrary, there are people in our research group concurrently working on protocol simulations. This is complementary work that will allow us to compare theoretical and empirical results, and will broaden our insight into the problems we are confronted to, giving us the ability to propose more comprehensive solutions. implementation.tex

6.1 Experimental Platform

In this section we will describe with sufficient details the setup of our experimental platform; this comprises the physical environment, the type of hardware and its configuration,

the software tools, and other elements that played an important role in our experiments. It is worth noting that we will describe the basic platform, as the configuration could be modified according to the particular experiment that should be carried out. When a specific configuration of the platform is used, we will proceed to describe it.

6.1.1 Hardware configuration

We used the following hardware for our test-bed:

- Lucent Wavelan IEEE 802.11 cards that allow for (nominal) throughputs of 4 and 6 Mpbs.
- Lucent Wavelan IEEE 802.11b cards that allow for (nominal) throughputs of 11 Mpbs. They are not compatible with the older IEEE 802.11 models, so they intercommunicate at rates that are common to both, that is, 1 or 2 Mbps.
- Lucent Wavepoint access points, which can use either the old or new card models. These access points are enhanced by an external antenna that increase the carrier's power by 17%, theoretically.
- A Pentium III 450 MHz desktop computer, running RedHat Linux and acting as a border router.
- One Pentium II 200 MHz and one Pentium II 350 MHz desktop computers running FreeBSD and acting as edge routers.
- Two Compaq Presario portable computers, AMD K6 350 MHz, running FreeBSD.
- Networking equipment, such as a 10 Mbps Ethernet hub, an Ethernet switch, etc.

6.1.2 Software environment

All our test machines were running free operating systems; this is not a random choice, since we needed access to the source code to study and modify as much as possible networking functionality:

- Desktop computers were running RedHat Linux.
- Portable computers were running FreeBSD.

Our work has been carried out in the context of the @IRS project [1], a French national project whose aim is the development of a testbed and protocols for the next-generation Internet. The IP protocol suite chosen for @IRS is Musica IP, a commercial product developed by 6Wind[1]. Musica is a dual-stack IPv4/IPv6 protocol suite implementing IP

[1] http://www.6wind.com

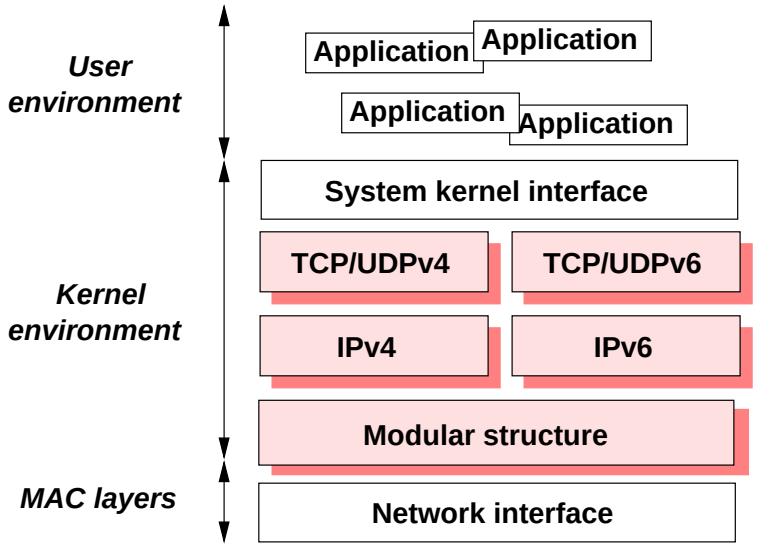

Figure 6.1: Architecture of the Musica IP suite.

and its associated protocols (ICMP, TCP, UDP, etc.). There are currently versions of the Musica suite for the Windows NT, and FreeBSD platforms. It should be noted that we did not have the source code available, and had only very basic documentation for it.

We also used several network test tools, some of them are standard tools that can be freely downloaded, some others were developed in-house to suit our needs:

- `wlmon`: a graphical tool, developed in-house, that allows monitoring of the quality of the signal; such tool was developed using the GTK graphical toolkit.

- `netperf`: a standard network tool that allows to determine the maximum throughput that a network can support with TCP traffic.

- `udpclient`: a simple request-response tool that sends UDP packets, at user-defined intervals, to a server that responds with ACK packets.

- `ping`: the standard Unix utility that shows round-trip delays of messages from the local machine to a remote one. This utility was also modified (and called `turboping`) to allow to send messages every 10 milliseconds.

- `ethereal`: a graphical network monitoring program that allows to observe on-going traffic in a chosen network interface.

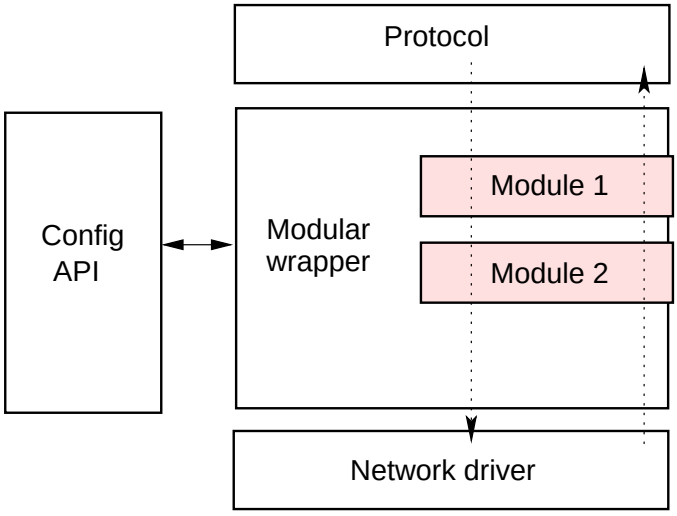

Figure 6.2: Musica's modular wrapper architecture.

6.2 QoS Management

6.2.1 Implementation of QoS mechanisms

One interesting feature of Musica, the TCP/IP stack we used, is that it allows the introduction of software modules (known as KLM or kernel loadable modules in the context of FreeBSD, the platform we used) that may modify the packets as they traverse the IP layer. Musica's architecture is shown in Figure 6.1.

Musica's modules are integrated in a modular wrapper that follows the *Bumped In The Stack* (BITS) approach, that consists of implementing the data processing functions (QoS, security, compression, etc.) between the TCP/IP stack and the hardware driver, the hardware being a networking card or a modem. This architecture presents the advantages of being transparent to applications and to the connectivity hardware being used. The architecture also implements an API for configuration of the modules. Figure 6.2 shows this modular architecture with two modules loaded into the wrapper.

Our group has developed Musica loadable modules[2] in order to implement the QoS mechanisms described in section 5.2. Functions belonging to core and edge routers are integrated into the same module. The QoS management modules are loaded into the mobile hosts and the nodes acting as access routers. These way traffic is constrained (*i.e.*, scheduled, shaped, etc.) before being sent to the wireless medium.

As shown in Figure 6.3, applications mark the flowlabel field[3] of their IP packets to indicate the QoS class they belong to (*i.e.*, GS, AS, BE). Once marked, the packets traverse the IP stack where the QoS module puts them into the appropriate class queue and dispatches them accordingly. As we will see ahead, we are currently working on the

[2]These are based on early versions of modules developed at LIP6 (`http://www.lip6.fr`).

[3]As Diffserv works on flow aggregates and not individual flows, the flowlabel field is normally unused.

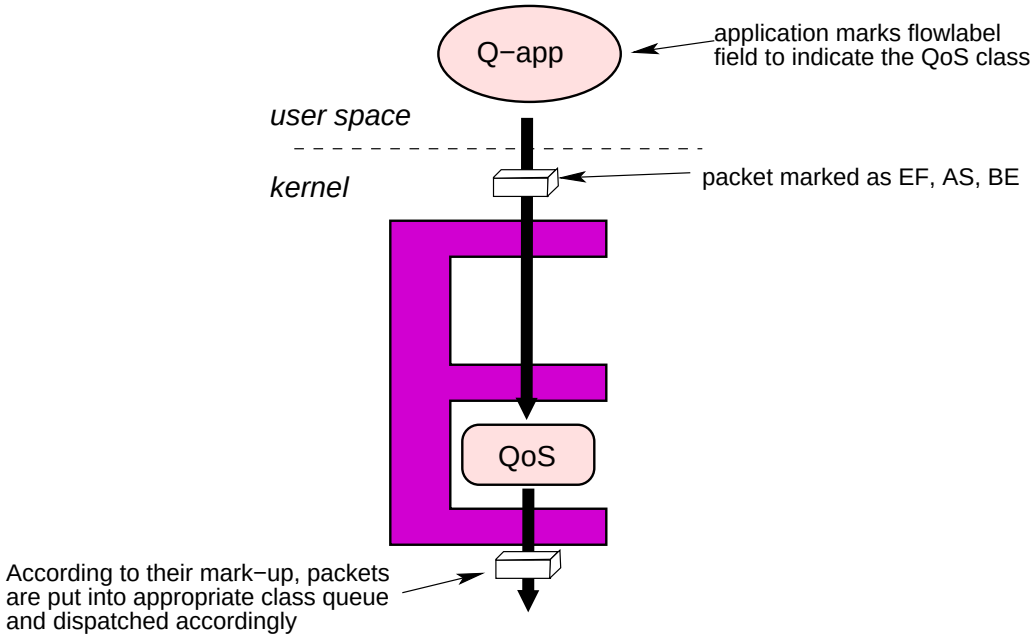

Figure 6.3: QoS support via a kernel loadable module.

development of a module that will mark the packets automatically by means of some pre-defined criteria. This way, it will not be strictly necessary that applications mark their packets.

The core and edge router behaviors are configured separately by means of two user-space configuration programs named `bboneconf` for the core and `edgeconf` for the edge behavior.

6.2.2 Experiments using our QoS mechanisms

We have conducted experiments in order to verify the performance of service differentiation and how different traffic classes are isolated. The setup for the experiments is depicted in Figure 6.4. A mobile host has two traffic sources: an UDP source generating traffic with a rate of 300 Kb/s with short 50 bytes packets (generated by the `udpclient` application mentioned earlier) and a TCP source generating an elastic traffic (`netperf` tool for measuring useful bandwidth with 1KB packets). In the first experiment, the QoS control mechanisms are inactive. Figure 6.5 presents the bandwidth obtained by the TCP source measured at the application layer. This is a greedy source that tries to get as much bandwith as possible —we can see that its bandwidth stays around 5 Mb/s— and is in competition with the UDP source that sends a fixed amount of traffic. We also show (Figure 6.6) the round trip delay (RTT) of the UDP traffic. Until `SeqNum = 100` both traffic sources are in competition and, as can be observed, the RTT of the UDP source is severely disturbed by the greedy TCP traffic, because both sources are scheduled according to the FIFO policy. Right after `SeqNum = 100`, the TCP source stops sending, so

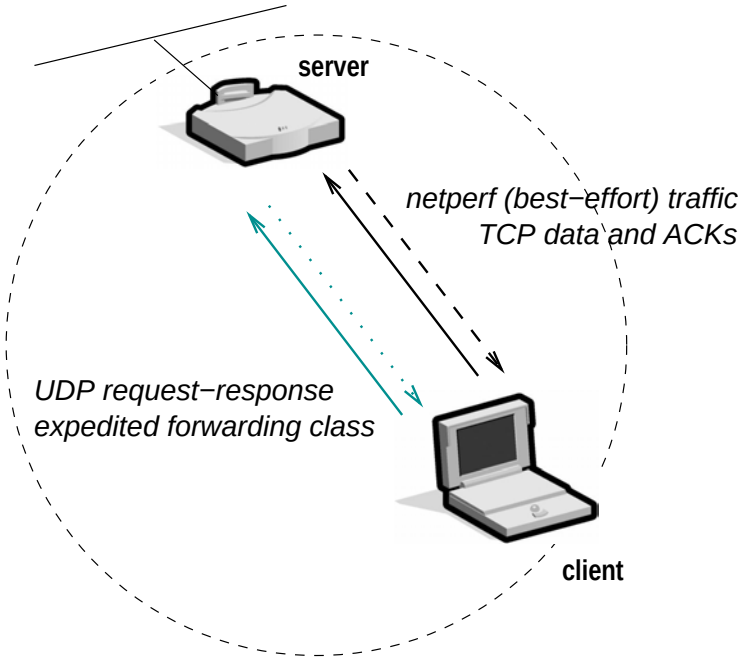

Figure 6.4: Experimentation set up.

that the RTT of the becomes shorter, around 2.5 ms, and much more predictable.

The second experiment tests the isolation of traffic classes by means of the *DiffServ* control mechanisms. The UDP source is assigned to the GS service class (EF behavior) and the TCP source is treated as Best-Effort (BE) traffic. The output traffic shaper is configured to limit the bandwidth of the BE class to 2.4 Mb/s. Figure 6.7 shows the bandwidth obtained by the BE source, which is effectively maintained around 2.4 Mb/s. It can also be seen that the RTT of the EF class is much less disturbed by the BE class (Figure 6.8). It is still greater than 2.5 ms, because of the competition with the BE class (the priority policy is not preemptive and an EF packet may wait an interval corresponding to the residual waiting time). As in the previous experiment, both traffic

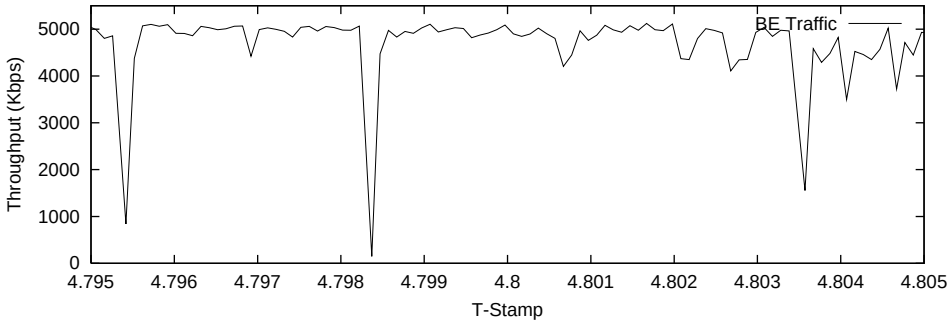

Figure 6.5: No QoS control, bandwidth of greedy TCP traffic.

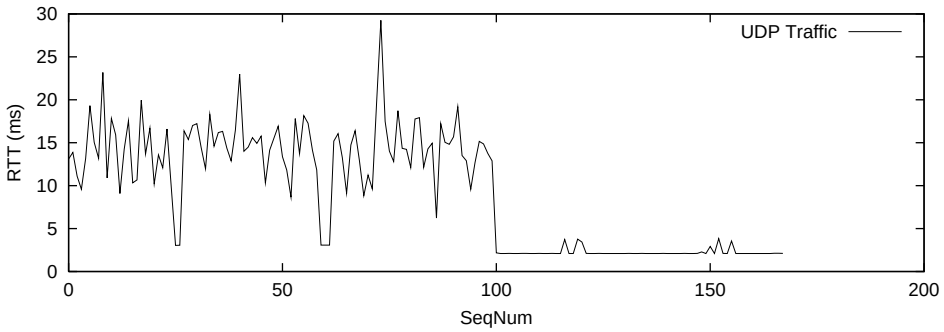

Figure 6.6: No QoS control, RTT of UDP traffic.

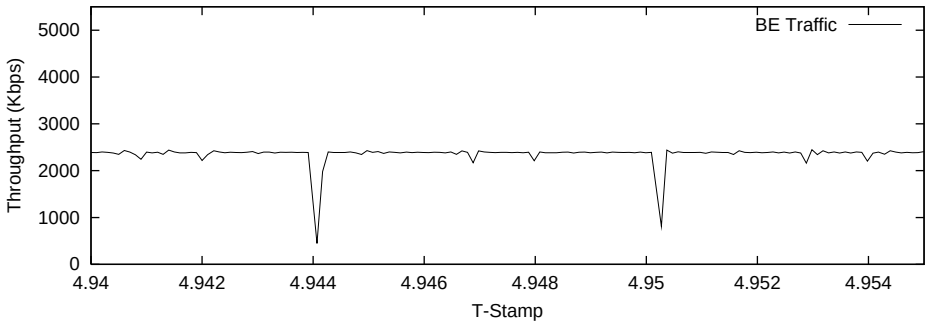

Figure 6.7: QoS control, bandwidth of BE class.

sources are competing for the shared medium but at a certain point (`SeqNum = 130`) the BE source stops sending. As it no longer has competition, the RTT of the EF traffic goes down to around 2.5 ms. These measures show that it is possible to isolate different QoS classes and obtain satisfactory performance.

The cohabitation of different QoS classes was further tested sending traffic from three sources. Initially a BE source is sent alone (Figure 6.9) which is effectively constrained around 2400 Kbps; then an AF source is sent concurrently (Figure 6.10). Around the instant 900 of that graphic an EF source is sent concurrently. From there up to instant 1800 –when the BE source is stopped– the RTT is perturbed, as the EF traffic has a higher priority, and the BE traffic tries to get the remaining bandwidth. The RTT goes down to around 4 ms when only the EF and AF flows are sharing the medium. Figure 6.11 shows the RTT of the EF flow. From instants 0 to 1200 the three flows are together, then the BE flow is stopped and the EF and EF flows share the channel up to instant 1600 when EF is alone and its RTT drops down to around 4 ms.

We also tested the performance of our QoS mechanisms when different machines send traffic within the same cell. One of the machines sends a greedy best-effort traffic and at the same time another machine sends a prioritary EF traffic. As Figure 6.12 shows, the BE traffic is effectively limited around 2400 Kbps. At the same time (Figure 6.13) the RTT of the competing EF traffic oscillates around 5 ms, with peaks of up to 12 ms; then

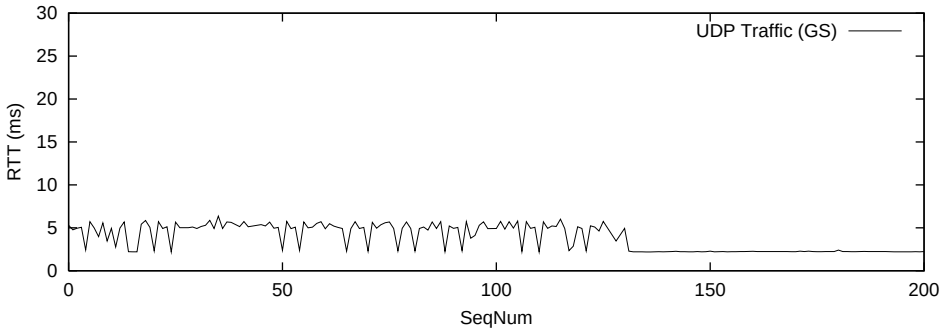

Figure 6.8: QoS control, RTT of EF class.

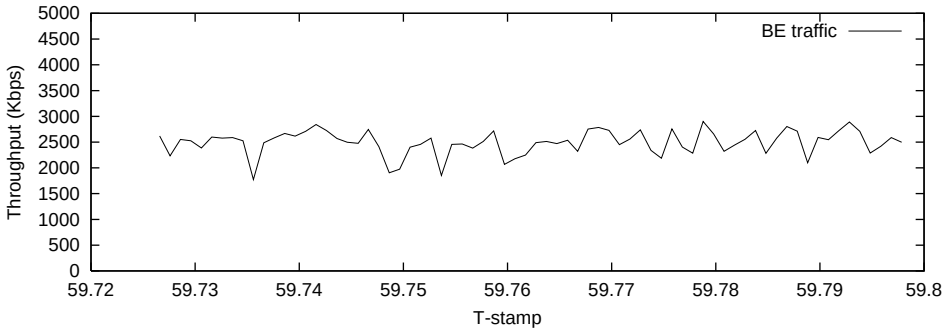

Figure 6.9: QoS control, bandwidth of BE class.

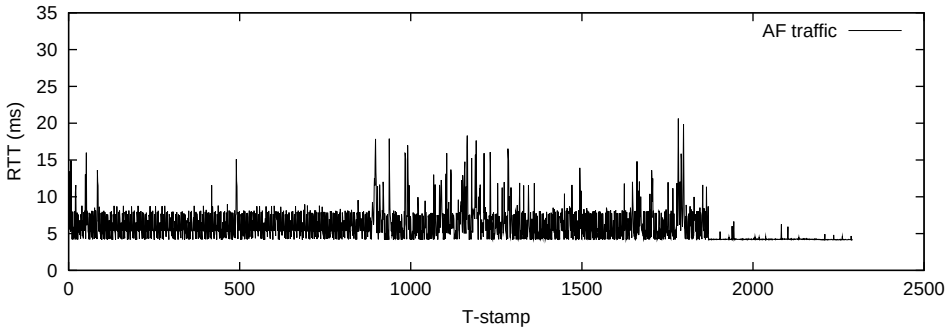

Figure 6.10: QoS control, RTT of AF class.

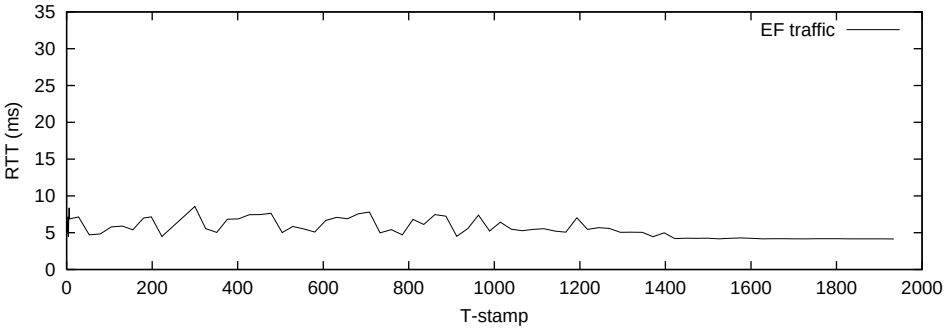

Figure 6.11: QoS control, RTT of EF class.

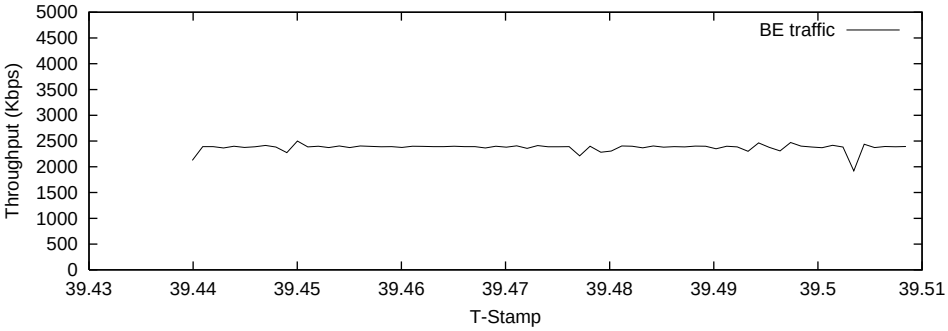

Figure 6.12: QoS control, bandwidth of BE class.

the BE traffic is stopped (`SeqNum=780`) and the RTT goes down staying steady around 4 ms.

6.3 Mobility Management

In Section 5.3 we introduced our proposed solutions for mobility management; we showed an architecture for mobility, gave details on the role of different elements and their interactions, and described a protocol for handling mobility of wireless hosts. We decided to prototype this architecture in a typical wireless environment, such as the one we have. What follows is the description of the development of the prototype. We also present problems we found along the way, limitations due to such problems, and the results we obtained.

The platform for setting up and testing our prototype for mobility management is shown in Figure 6.14. It is composed of an edge router (`djerba`), two access routers (`grenade` and `gomera`), and a mobile host (`milos`). Their IP addresses and network prefixes are also shown. The hand-off protocol has been prototyped in both IPv4 and IPv6, although for the IPv6 there are still some pending issues that we will discuss later. Hence, the following discussion describes the IPv4 version of the prototype.

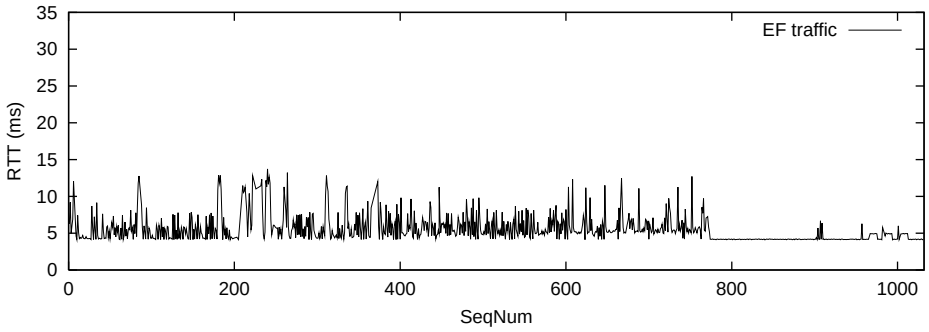

Figure 6.13: QoS control, RTT of EF class.

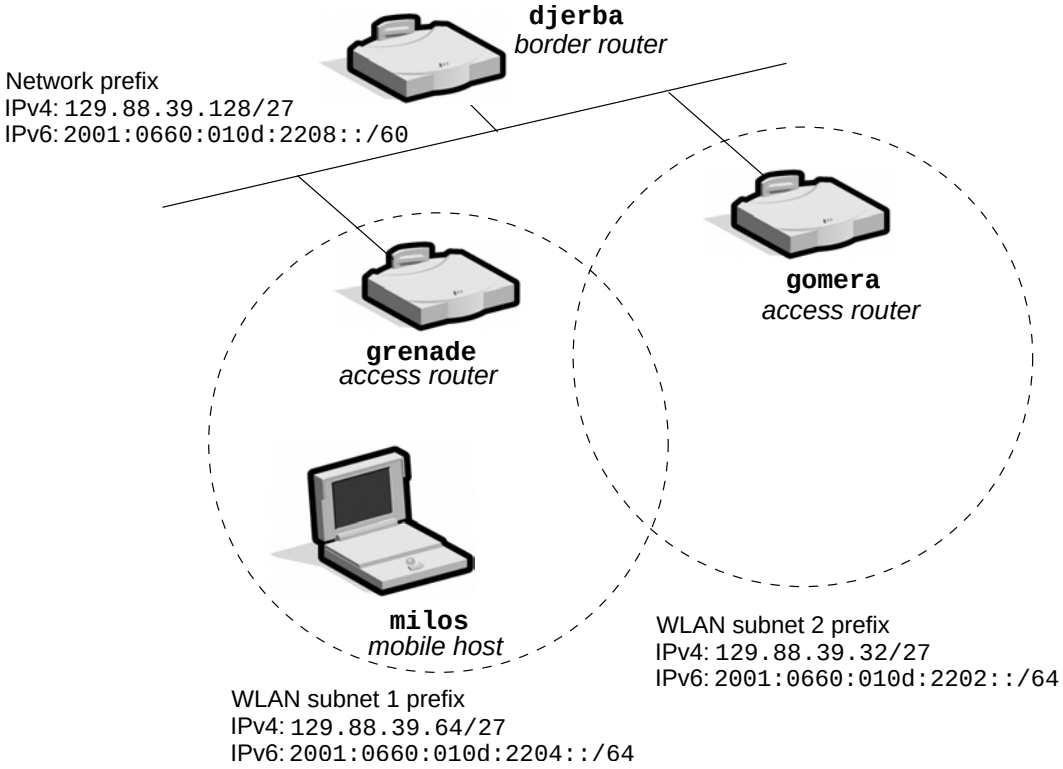

Figure 6.14: Platform setup for testing mobility management.

6.3.1 Layer 2 control

Having tested the performance of our wireless platform (*cf.* Section 5.1), we also wanted to know to what degree it was possible to control the layer 2 functionality of our available equipment (*i.e.*, Lucent's WaveLAN cards and WavePoint base stations). For the case of base stations, we were either interested in:

1. having a workstation, acting as access router, to be directly linked to the base station to control its functionality; this includes requesting how many mobile hosts were currently in its cell, sending inter-access-point protocol commands, etc.
2. have a WaveLan-equipped workstation to emulate the functionality of an access point.

Both cases suppose the availability of technical specifications for the access points, which was not true for us. Due to their commercial interests, Lucent does not provide open access to their equipment's functional specifications. For the case of controlling the wireless networking cards, the situation was not much better, as the only documents made available by Lucent [153; 154] were very incomplete and did not provide enough in-depth technical details. We needed in particular the capability to control the card to enter the promiscuous mode (also known as spy mode) to be able to monitor signals coming from different stations, and then instruct it to handoff to one of them. As evidenced by results from independent developers who reverse-engineered Lucent's equipment [150; 3] and by our own group efforts, this type of control is not possible (at least not without having the proper documentation). We have, thus, very limited control abilities:

- Signal monitoring for neighboring cells does not work in the BSS (infrastructure) mode. This monitoring can be done in ad-hoc mode, however
- the ad-hoc mode does not allow changing channels, so, a handoff is only possible if all involved cells use the same channel.

6.3.2 Protocol implementation

We used Lucent's WaveLAN cards configured for operation in the ad-hoc mode which, as explained earlier (Section 6.3.1), is the only mode that allowed to obtain signal to noise measurements in neighbor cells. However, this mode does not allow changing channels, so neighbor cells had to use the same communication channel. A side effect of this limitation is that the prototyped handoff protocol was simpler, because both access routers were able to communicate directly with each other. Thus, we decided to incorporate some optimizations so that when the mobile sends a handoff request, the current access router passes the request to the target access router directly, without going through the cross-over router.

The mobile host was obviously using the same frequency, so it was able to listen to neighbor access routers simultaneously, meaning not only that at handoff completion the

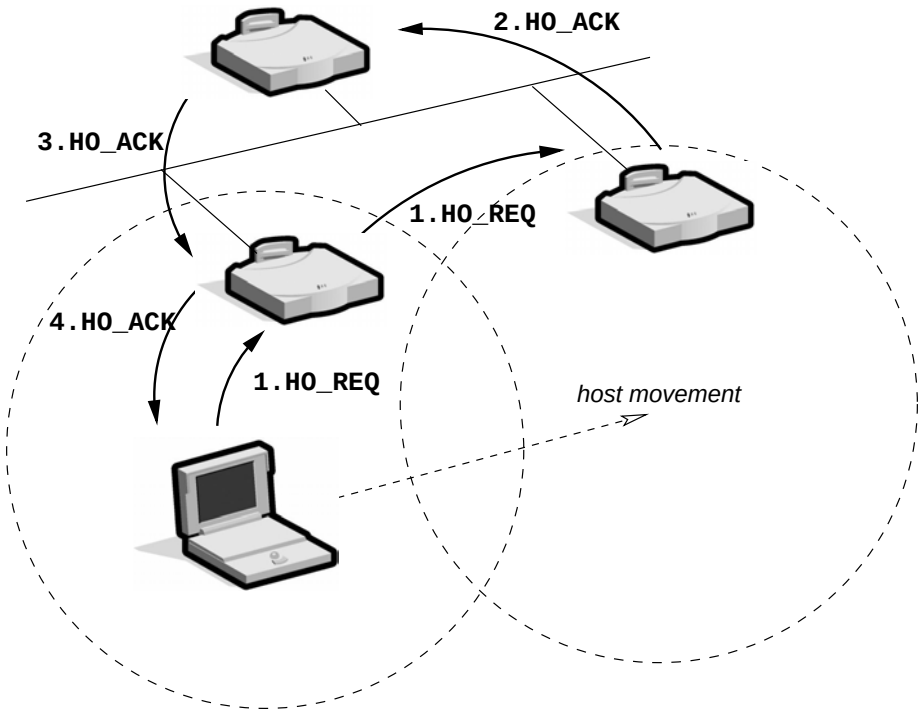

Figure 6.15: Handoff execution.

target access router was able to send an ACK directly to the mobile host, but that during the time that the new path was being setup, the host was able to communicate with both access routers, making unnecessary the use of the dual-cast mode (which, for this reason, we have not currently implemented). Obviously, using the same frequency for contiguous cells is not desirable in general, because we want to provide sufficient bandwidth to better guarantee QoS.

Our current prototype of the handoff protocol has been implemented in user space using UDP, although we have plans to move the implementation to kernel space (possibly using ICMP) expecting further performance enhancements. Currently, mobility daemons executing on routers wait for handoff messages and perform route updates as requested. These daemons have the capability of accessing the routing tables to modify or add entries to them (using `socket PF_ROUTE` data structures); for this reason the daemons have to run in privileged (*i.e.*, root) user mode.

6.3.3 Experiments and results

The following is a typical experiment (see Figure 6.15) we conducted in order to verify the correctness of our prototype. The mobile host (`milos`) has `grenade` as its current access router, and it will try to handoff to a neighboring cell, taking `gomera` as its new access router. What happens is the following:

1. `milos` starts moving out to the periphery of its current cell towards `gomera`'s cell. When the S/N ratio becomes too low, a handoff request (`HO_REQ`) destined to `gomera` is sent. The request passes through `grenade`, that forwards it directly to `gomera`.

2. `gomera` receives the request and adds a host route to its table indicating that all traffic for `milos` should be forwarded via the wireless interface. Then, a message (`HO_ACK`) is sent to `djerba`.

3. Analogously, `djerba` adds a host route to its table indicating that traffic for `milos` should be directed to `gomera` and sends `HO_ACK` to `grenade`.

4. `grenade` sends a `HO_ACK` to `milos` and changes its table to sent traffic for `milos` via `djerba`.

5. When `milos` receives the `HO_ACK` it means that the handoff has been successfully completed, so it modifies its router table to make `gomera` its default next hop router.

During the course of experiments like the one described above, we also made some measurements in order to determine packet loss caused by handoff and total time for handoff. To do this, `milos` executed the `ping` command to send traffic to a host (such as `delos`) behind `djerba`.

In the first series of experiments we used `ping` with the `-R` (record route) option that shows the route followed by packets. By default, `ping` sends a packet once a second.

```
milos-wl:~: ping -R delos
1-  PING delos.imag.fr (129.88.38.94): 56 data bytes.
2-  64 bytes from milos.imag.fr (129.88.38.94): icmp_seq=0 ttl=253 time=15.400 msec
3-  RR:  grenade-39.imag.fr (129.88.39.151)
4- djerba.imag.fr (129.88.38.149)
5- delos.imag.fr (129.88.38.94)
6- delos.imag.fr (129.88.38.94)
7- djerba.imag.fr (129.88.39.158)
8- grenade-wl.imag.fr (129.88.39.94)
9- milos-wl.imag.fr (129.88.39.65)
10- 64 bytes from 129.88.38.94: icmp_seq=1 ttl=253 time=21.768 ms (same route)
11- 64 bytes from 129.88.38.94: icmp_seq=2 ttl=253 time=30.179 ms (same route)
12- 64 bytes from 129.88.38.94: icmp_seq=3 ttl=253 time=6.564 ms  (same route)
13- 64 bytes from 129.88.38.94: icmp_seq=4 ttl=253 time=9.531 ms  (same route)
14- 64 bytes from 129.88.38.94: icmp_seq=5 ttl=253 time=10.539 ms (same route)
15- 64 bytes from 129.88.38.94: icmp_seq=6 ttl=253 time=4.885 ms  (same route)
16- 64 bytes from 129.88.38.94: icmp_seq=7 ttl=253 time=4.894 ms  (same route)
17- 64 bytes from 129.88.38.94: icmp_seq=8 ttl=253 time=4.901 ms  (same route)
18- 64 bytes from 129.88.38.94: icmp_seq=9 ttl=253 time=6.484 ms  (same route)
19- 64 bytes from 129.88.38.94: icmp_seq=10 ttl=253 time=9.639 ms (same route)
20- 64 bytes from 129.88.38.94: icmp_seq=11 ttl=252 time=13.328 ms
21- RR:  gomera.imag.fr (129.88.39.145)
22- djerba.imag.fr (129.88.38.149)
23- delos.imag.fr (129.88.38.94)
24- delos.imag.fr (129.88.38.94)
25- djerba.imag.fr (129.88.39.158)
26- grenade-wl.imag.fr (129.88.39.94)
27- gomera-wl.imag.fr (129.88.39.62)
28- milos-wl.imag.fr (129.88.39.65)
29- 64 bytes from 129.88.38.94: icmp_seq=12 ttl=252 time=13.431 ms
30- RR:  gomera.imag.fr (129.88.39.145)
```

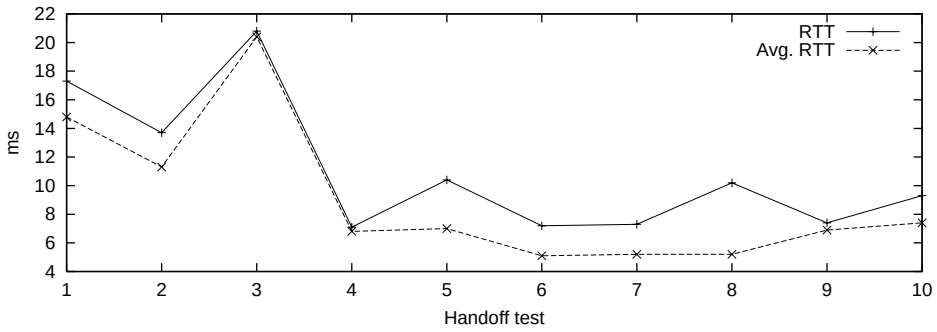

Figure 6.16: RTT and average RTT during handoff.

```
31- djerba.imag.fr (129.88.38.149)
32- delos.imag.fr (129.88.38.94)
33- delos.imag.fr (129.88.38.94)
34- djerba.imag.fr (129.88.39.158)
35- grenade-39.imag.fr (129.88.39.151)
36- gomera-wl.imag.fr (129.88.39.62)
37- milos-wl.imag.fr (129.88.39.65)
38- 64 bytes from 129.88.38.94: icmp_seq=13 ttl=252 time=10.755 ms (same route)
39- 64 bytes from 129.88.38.94: icmp_seq=14 ttl=252 time=13.205 ms (same route)
40- 64 bytes from 129.88.38.94: icmp_seq=15 ttl=252 time=4.951 ms  (same route)
41- 64 bytes from 129.88.38.94: icmp_seq=16 ttl=252 time=41.285 ms (same route)
42- 64 bytes from 129.88.38.94: icmp_seq=17 ttl=252 time=4.964 ms  (same route)
43- 64 bytes from 129.88.38.94: icmp_seq=18 ttl=252 time=5.741 ms  (same route)
44- 64 bytes from 129.88.38.94: icmp_seq=19 ttl=252 time=4.907 ms  (same route)
45- 64 bytes from 129.88.38.94: icmp_seq=20 ttl=252 time=14.064 ms (same route)
46- 64 bytes from 129.88.38.94: icmp_seq=21 ttl=252 time=8.123 ms  (same route)
47- 64 bytes from 129.88.38.94: icmp_seq=22 ttl=252 time=4.999 ms  (same route)
48- 64 bytes from 129.88.38.94: icmp_seq=23 ttl=252 time=4.909 ms  (same route)
^C
--- delos.imag.fr ping statistics ---
24 packets transmitted, 24 packets received, 0% packets loss
round-trip min/avg/max/stddev = 4.885/11.227/41.285/8.708 ms
```

In the screen-dump above, we added line numbers to make the description of what happened easier. As can be seen, at line 21 a route change is made, although packets still pass through `grenade` on their way to `gomera` since `djerba`'s routing table has not been changed; no packets are lost during the process. It is noticeable how the roundtrip times (RTT) have large variations, but at the time of handoff the RTT is well within the standard variation of the average value.

We performed this experiment ten times in a row, and we present (in Figure 6.16) the resulting round-trip times at the time of handoff, as well as the average round-trip time. No packet loss occurred during handoff (as evidenced by the screen-dump above), so we do not include this factor in the graphic.

A second series of experiments, aimed to have better insight into packet losses, consisted of using `ping` with the -f option that allows packets to be sent one hundred times a second, or as fast as possible. In Figure 6.17 we can see that although the number of packets lost is low, some packets were lost during handoff.

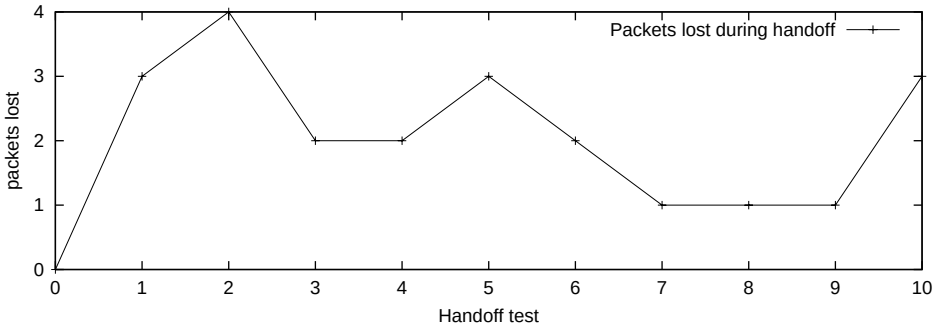

Figure 6.17: Packet loss during handoff.

We conducted another series of experiments to measure with greater accuracy the total delay of handoff. We used `ping` with the `-f` option and the `tcpdump` network tool. In these experiments, besides using `ping -f` as done earlier, we run `tcpdump` on `milos` to capture the activity of the wireless. The next screendump shows what happened.

```
milos-wl:~: tcpdump wi0
tcpdump: listening on wi0
1-  16:31:06.364108 milos-wl.imag.fr > delos.imag.fr: [icmp]
2-  16:31:06.369645 delos.imag.fr >  milos-wl.imag.fr: [icmp]
3-  16:31:06.688797 milos-wl.imag.fr.5232 >  grenade-wl.imag.fr.5232: udp 16
4-  16:31:06.691981 grenade-wl.imag.fr.5232 > milos-wl.imag.fr.5232: udp 20
5-  16:31:07.376991 milos-wl.imag.fr > delos.imag.fr: [icmp]
6-  16:31:07.381508 delos.imag.fr >  milos-wl.imag.fr: [icmp]
7-  16:31:08.384134 milos-wl.imag.fr > delos.imag.fr: [icmp]
8-  16:31:08.388543 delos.imag.fr >  milos-wl.imag.fr: [icmp]
9-  16:31:09.394199 milos-wl.imag.fr > delos.imag.fr: [icmp]
10- 16:31:09.398280 delos.imag.fr >  milos-wl.imag.fr: [icmp]
11- 16:31:10.404175 milos-wl.imag.fr > delos.imag.fr: [icmp]
12- 16:31:10.408285 delos.imag.fr >  milos-wl.imag.fr: [icmp]
```

As can be seen, `milos` is initially sending packets to `delos` while linked to `gomera` via the wireless link. Then, at step 3, a packet for handoff request is sent to `grenade`. At step 4, the packet that acknowledges the handoff is sent to `milos`. So, as far as `milos` is concerned, the handoff has been completed. Handoff latency, as seen from steps 3 and 4, is around 3.184 ms. The experiment was repeated ten times, and resulting times are given in Figure 6.18.

The average handoff latency is around 3.3624 ms. However, this is only the time elapsed between the handoff request and reception of the acknowledgment. It should also be taken into account the time it will take `milos` to process the acknowledgment, including modifying its routing table to change its default access router. For this purpose, we used the `time` command, which in BSD has a resolution of 10 ms. So, 10 ms can be considered as an upper limit for this delay. Our measured hand-off latency is fairly low compared to the performance of Mobile IP [63; 101].

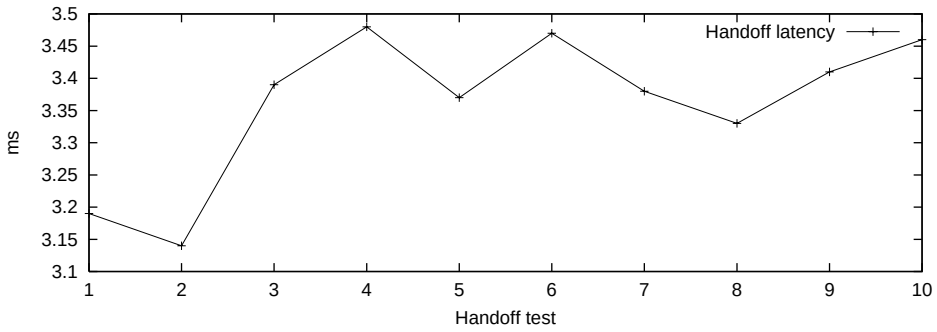

Figure 6.18: Handoff latency.

6.3.4 Pending issues

The current prototype that implements our mobility architecture has allowed us to validate our proposals and to give us an idea of the performance that can be achieved by on an operational platform, not only by using simulation software. Although the results we have obtained up to now are very satisfactory, there are some pending issues that we are currently addressing. We will discuss them below.

The experiments we presented above were conducted using the IPv4 version of our prototype. The reason for this is that, although we already have an IPv6 version[4], we have experienced problems due to IPv6 auto-configuration. For instance, a little while after a daemon modifies a routing table for new path setup, an IPv6 router sends a `router advertisement` message that re-establishes a previously deleted entry, having then a situation where both the old and new entries are present. This becomes particularly annoying in the case of a mobile host that finds itself having two default next-hop routers (*i.e.*, the old and new access routers).

Using the wireless cards in the ad-hoc mode, where all wireless nodes use the same frequency channel, is not an ideal situation. We were forced to momentarily use this mode of operation because of the problems (notably lack of documentation) to control the available hardware's functions. We have recently acquired new (in theory more open) wireless hardware [12], with which we may be able to operate in infrastructure mode, where contiguous cells will use different channels. We will also be able to have more MAC level control, such as the ability to monitor the signal/noise ratio in all operating modes, changing the current channel (*e.g.*, for handoffs), etc.

Finally, we have implemented the main features of our proposed protocol, that allow us to verify its usefulness and performance, but have purposely left for a next stage the implementation of other features that will contribute to a more complete and robust prototype. Among these features we can mention: the ability to perform a rollback in case of router failure during a handoff procedure and the dual-cast feature for the case when the mobile host can not simultaneously listen to the old and new access routers.

[4]In fact, there's only one coded prototype that takes advantage of using a dual processing IPv4/v6 stack.

6.4 Integrated Architecture

As we have seen, the basic features of our proposal for the management of mobility and QoS have already been prototyped and experiments have been conducted that show encouraging results. Nevertheless, we are currently working on the implementation of other features that will allow us to have a more complete prototype that encompasses all aspects of the integrated architecture we envision (*cf.* Section 5.4.3), notably, reverse paging and in-band signaling. We will now discuss the status of this preliminary work.

6.4.1 Reverse Paging

As stated in Section 5.4.1, reverse paging is an extension to the mobility management protocol we have already implemented. Given that our current prototype uses UDP for sending protocol commands, then, location advertisement and other reverse paging messages are also being implemented this way. But there are other problems harder to solve, and requiring further investigation for their implementation.

For the case when a mobile host moves while off or in dormant mode and wakes up at another cell, radio communication (layer 2) is handled by the 802.11-compliant equipment. But at layer 3 (IP) the situation is different, since the mobile will not know the address of its new next hop access router to modify its own routing table accordingly. We are exploring two possibilities to solve this:

1. After it wakes up, the mobile host will beacon broadcast messages to advertise its presence. When the new access router within range receives the advertisement, it will start the usual handoff procedure, adding the mobile to its routing table, and finally sending a `HO_ACK` so the mobile can make the new access router its default access router.

2. Since we are in the context of micro-mobility, the area of motion is restricted and thus the number of potential new access routers is limited. The mobile could have a list of routers' addresses, starting with the neighboring ones, that it could traverse adding to its routing table and trying to establish contact with. When the right one responds, the usual handoff procedure is initiated.

6.4.2 Integration via in-band signaling

Going back to Section 5.4.3, we can see that our proposed integrated architecture couples in-band and out-of-band protocols for QoS and mobility management. The implementation of some of the elements of the architecture has already been carried out and has been discussed. Nonetheless, some key entities still need to be fully implemented. This is the case for the packet classifier and the QoS manager.

Although the mechanisms that provide QoS support have been implemented and tested, they are currently configured manually, basically by editing configuration files.

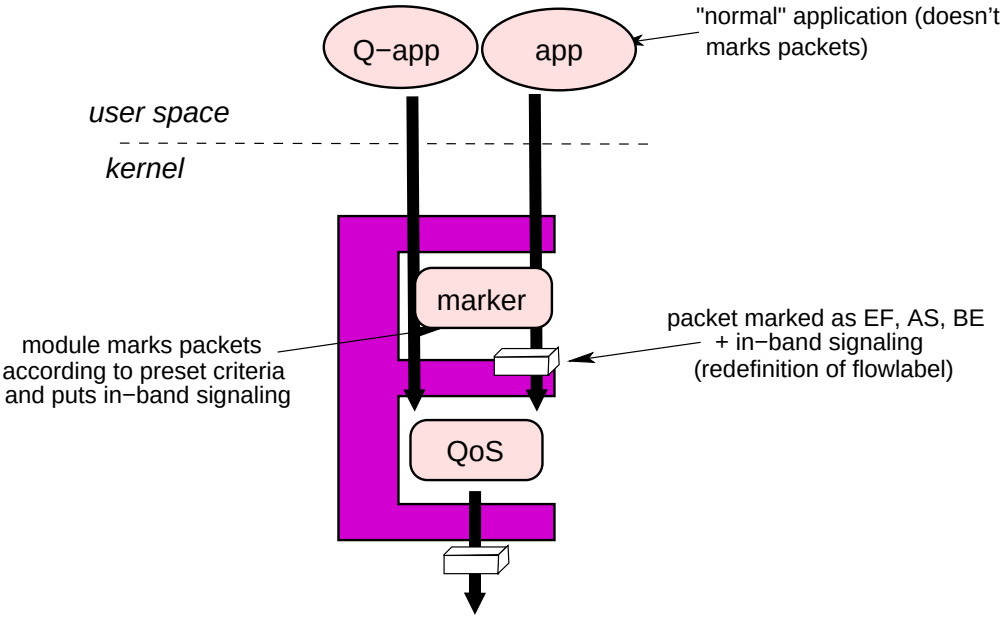

Figure 6.19: Role of the new kernel module towards an integrated architecture.

Also, it is currently the case that applications should mark their IP packets to indicate what class they belong to, allowing the QoS mechanisms to react accordingly. We have started the development of the QoS monitor and the packet classifier, coding functions pertaining to both into a single kernel loadable module. This way, the IPv6 flowlabel header field is used to mark packets according to configured criteria, but also to insert in-band protocol commands. Recall that, as shown in Figure 6.3, a module for QoS mechanisms was already in place. Now, before going through that module, packets pass through the module currently under development to be marked for classification and to carry the corresponding in-band information. This can be seen in Figure 6.19.

6.5 Chapter conclusions

In this chapter we have tested the functionality, performance, and applicability of the solutions we proposed in Chapter 5. We took an empirical approach by developing working prototypes that are the foundation of an evolving test platform. Our QoS mechanisms were implemented in a kernel loadable module. We conducted a series of tests and observed that traffic for different classes is isolated and the delay is stabilized. The basic mechanisms for our mobility management proposal are currently implemented in user-space daemons. The mobility tests we performed showed a good performance of the protocol. We have continued developing the building blocks of or proposed integrated architecture and some elements are still work in progress.

Conclusions

Chapter 7

Conclusions

Mistakes are the portals of discovery.

— James Joyce

7.1 Achievements

An important current trend is the use of wireless local area networks (WLANs) as "hot-spots" for high-speed Internet access. Although WLAN technologies are achieving ever higher transmission speeds (with commercial products of up to 54 Mbps already announced), there is however a noticeable lack of solutions encompassing the provision of QoS and mobility. Layer 3 micro-mobility solutions have been proposed, but they either completely ignore or superficially address the QoS provision aspect. Thus, efforts for handling mobility in IP networks and for providing a better than best effort QoS are being addressed, but they constitute independent efforts that usually have no correlation.

We have conducted performance tests that gave us insight into the actual characteristics of WLANs. According to the results of these tests, we were able to propose mandatory pre-conditions for providing QoS guarantees on a WLAN platform. Our proposal for QoS management leverages existing work in the area of IP QoS by adopting the Diffserv model proposed by the IETF. Given the performance instabilities that WLANs show in time and space, it is not reasonable to try to adopt a QoS model that offers hard guarantees. The Diffserv model we have adopted fits a lot better the variability and dynamic nature of WLANs, providing statistical guarantees for the provision of QoS. We have proposed the use of traffic controls to allow bandwidth allocations for different traffic sources. Our experimental results show that it is possible to isolate the traffic for different classes of service, providing substantially better performance to higher priority traffic.

The approach we have taken in our work is toward the integrated management of mobility and QoS within IP-based WLANs. We have proposed a mobility management protocol that is simple and efficient. Simplicity was a design feature necessary not only to achieve a performance level adequate for the provision of QoS, but also needed for

efficiently handling mobility in local environments. Even the order in which the handoff protocol messages traverse the nodes has been designed taking into account the eventual incorporation of QoS-signaling. Our proposed architecture for the integrated management of mobility and QoS includes a light-weight in-band signaling protocol and (what we have termed) reverse paging.

7.2 Perspectives

We have proposed an integrated architecture for mobility and QoS management, and although several components of the architecture have been already implemented, the global implementation and integration is still work in progress.

Regarding our mobility management proposal, our current implementation has allowed us to show its correctness and performance in an actual test platform. However, handling some exceptional conditions have to be implemented into the prototype. The eventual implementation of MAC-level control features will allow the prototype to properly work in the BSS (infrastructure) mode. Also, some issues regarding IPv6-specific features, such as auto-configuration, have to be resolved for the IPv6 version of the prototype. Installation of Mobile IP in our test platform for handling global mobility, in order to complement our micro-mobility management, is also an activity we will carry out. We have a preliminary version of a kernel module that marks packets to allow their classification by our QoS mechanisms, and also inserts in-band signaling commands. However, the current criteria for marking packets is extremely simple and consists of looking at static information such as (source and destination) ports and addresses; more complex criteria should be incorporated, such as type of application, user profiles, etc. This implies the development of the configuration manager. Also, even though the basic functions for inserting in-band information into packets has already been coded, the criteria for defining when and what information should be inserted, has not. QoS monitor functions are currently in an early stage for development. For instance, only resources such as bandwidth and signal strength are taken into account. Also, the interaction with the mobility monitor (*e.g.*, to to inform it of current resources' status) is still lacking. Although a kernel module implementing QoS mechanisms is already in place, changes need to be made so it can accommodate our proposed semantics for the IPv6 flowlabel field. In its current state, the module will overwrite any information we put into it.

A great body of experience and important momentum has been built within our research group. We are confident that the pending implementation and integration issues constitute engineering tasks that will be easily surmounted in the immediate coming months. Recent presentations of our work's findings in peer-reviewed international conferences [53; 52] have generated interest and positive remarks that are very encouraging.

The research work presented in this document has also allowed us to define new exciting directions to explore. We have built an interesting platform that can constitute a good starting point to investigate seamless mobility in heterogeneous environments where different wireless networking technologies may be used; this implies going further in our proposals concerning handoff between cells, arising the need to perform context transfer

for preserving the capabilities, security levels and QoS guarantees when moving from cell to cell. We are also interested in complementing our network layer proposals with session layer approaches that provide abstractions to provide mobility transparency to higher layers, including applications.

"Don't let it end like this. Tell them I said something. "
— last words of Pancho Villa (1877-1923)

Bibliography

Bibliography

[1] @IRS project. Integrated Architecture for Networks and Services. `http://www-rp.lip6.fr/airs/`, 2001.

[2] I. Aad and C. Castelluccia. Differentiation mechanisms for IEEE 802.11. In *INFOCOM 2001*, 2001. Anchorage, Alaska.

[3] Absolute-Value. Linux WLAN Project. `http://www.linux-wlan.com/linux-wlan/`.

[4] S. Alexander and R. Droms. DHCP Options and BOOTP Vendor Extensions. `Internet RFC 2132`, 1997.

[5] P. Almquist. Type of Service in the Internet Protocol Suite. `Internet RFC 1349`, 1992.

[6] P. Anelli and G. Legrand. Differentiated services over shared media. In *Intl. Workshop on QoS, IWQoS 01*, 2001.

[7] E. Ayanoglu, K. Eng, and M. Karol. Wireless ATM: limits, challenges, and proposals. *IEEE Personal Communications Magazine*, 3(4), 1996.

[8] F. Baker. Requirements for IP Version 4 Routers. `Internet RFC 1812`, 1995.

[9] F. Baker, R. Guerin, and D. Kandlur. Specification of Commited Rate Quality of Service. `Internet Draft. draft-ietf-intserv-commit-rate-svc-00.txt`, 1996.

[10] J.-C. Bennet and H. Zhang. WF2Q: Worst-case Fair Weighted Fair Queueing. In *Proc. INFOCOM 96, San Francisco, CA*, pages 120–128, 1996.

[11] J. C. Bennett and H. Zhang. Hierarchical Packet Fair Queueing Algorithms. *IEEE/ACM Transactions on Networking*, 5(5):673–689, 1997.

[12] H. Bensaid and J. Poizat. Etude et adaptation d'un driver de carte PCMCIA compatible 802.11b. Rapport de projet de fin d'etudes, ENSIMAG, june 2001.

[13] P. Bhagwat, C. Perkins, and S. Tripathi. Network Layer Mobility: an Architecture and Survey. *IEEE Personal Communications Magazine*, 3(3):54–64, 1996.

[14] R. Bhat. Wireless atm requirement specification. ATM Forum, RTD-WATM-01.02.

[15] S. Bhatti and J. Crowcroft. QoS-Sensitive Flows: issues in IP packet handling. *IEEE Computer*, pages 48–57, july-august 2000.

[16] G. Bianchi and A. Campbell. A Programmable MAC. In *Proc. Intl. Conf. on Universal Personnal Communications, ICUP 98*, 1998. Florence, Italy.

[17] U. Black. *QoS in Wide Area Networks*, chapter 2. Prentice Hall, 2000.

[18] D. Blair et al. SeaMoby Micro-mobility Problem Statement. `Internet Draft draft-ietf-seamoby-mm-problem-01.txt`, work in progress, february 2001.

[19] S. Blake, D. Black, and M. Carlson. An Architecture for Differentiated Services. `Internet RFC 2475`, 1998.

[20] M. Blumenthal and D. Clark. Rethinking the design of the Internet: The end to end arguments vs. the brave new world. *ACM Trans. Internet Technology*, 2001. *To appear.*

[21] Bouyer and E. Horlait. Bandwidth Management and Reservation over Shared Media. In *SFBSID 97*, 1997. Fortaleza, Brazil.

[22] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture. `Internet RFC 1633`, 1994.

[23] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP)–Version 1 Functional Specification. `Internet RFC 2205`, 1997.

[24] R. Caceres and V. Padmanabhan. Fast and Scalable Handoffs for Wireless Internetworks. In *ACM Mobicom 96*, 1996.

[25] A. Campbell and J. Gomez-Castellanos. IP Micro-mobility Protocols. *ACM Sigmobile Mobile Computer and Communications Review*, 2001.

[26] A. Campbell et al. An Overview of Cellular IP. In *IEEE Wireless Communications and Networks Conference, WCNC*, pages 606–611, 1999.

[27] B. Carpenter. What's All the Fuss About Differentiated Services? *IEEE Computer*, pages 111–112, 1999.

[28] C. Castelluccia. A Hierarchical Mobile IPv6 Proposal. Technical Report INRIA RT-0226, 1998.

[29] I. Castineyra, J. Chiappa, and M. Steenstrup. The Nimrod Routing Architecture. `Internet RFC 1992`, 1996.

[30] J. Chan et al. The Challenges of Provisioning Real-Time Services in Wireless Internet. *Telecommunications Journal of Australia*, 50(3), 2000.

[31] S. Chesire and M. Baker. Internet Mobility 4x4. In *ACM SIGCOMM Computer Comm. Review*, pages 318–329, 1994.

[32] Cisco. DiffServ —The Scalable End-to-End QoS Model. (Search online at:) `http://www.cisco.com/`, whitepaper, 2001.

[33] M. Corson and A. Campbell. Toward Supporting Quality of Service in Mobile Ad hoc Networks. In *First IEEE OPENARCH'98, San Francisco, CA., USA*, 1998.

[34] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[35] R. Cruz. A calculus of network delay. Part 1: network elements in isolation. *IEEE Trans. on Information Theory*, 37:114–131, 1991.

[36] D. Trossen and G. Krishnamurthi and H. Chaskar. Issues in candidate access router discovery for seamless IP handoffs. `Internet Draft draft-ietf-seamoby-cardiscovery-issues-00.txt`, work in progress, july 2001.

[37] DARPA. DARPA Internet Program Protocol Specification. `Internet RFC 791`, 1981.

[38] DARPA. Transmission Control Protocol. `Internet RFC 793`, 1981.

[39] S. Deering. ICMP Router Discovery Messages. `Internet RFC 1256`, 1991.

[40] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6). `Internet RFC 1883`, 1995.

[41] Demers, Kenshav, and Shenker. Analysis and Simulation of a Fair Queuing Algorithm. *Internetwork: Research and Experience*, 1(1), 1990.

[42] M. V. der Zee. Quality of service in bluetooth networking. `http://ing.ctit.utwente.nl/WU4/Documents/`, 2000.

[43] G. Dommety et al. Fast Handovers for Mobile IPv6. `Internet Draft draft-ietf-mobileip-fast-mipv6-02.tx`, work in progress, 2001.

[44] R. Droms. Dynamic Host Configuration Protocol. `Internet RFC 2131`, 1997.

[45] K. Eng, M. Karol, , M. Veeraraghavan, E. Ayanoglu, C. Woodworth, and A. Valenzuela. A Wireless Broadband Ad-hoc ATM Local Area Network. *ACM/Baltzer Wireless Networks Journal*, 1(2):161–174, 1995.

[46] Ericsson. Ericsson demonstrates hiperlan 2 prototypes. press release, december 11, 2000. `http://www.ericsson.com/press/20001211-0067.html`.

[47] K. Fall. The NS Manual. `http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf`, 16 march 2001.

[48] P. Ferguson and G. Huston. *Quality of Service: Delivering Qos on the Internet and in Corporate Networks*, chapter 7. John Wiley and Sons, 1998.

[49] S. Floyd. Congestion Control Principles. `Internet RFC 2914`, 2000.

[50] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[51] F. Fluckiger. *Understanding Networked Multimedia.* Prentice Hall, 1995.

[52] J.-A. García-Macías, F. Rousseau, G. Berger-Sabbatel, L. Toumi, and A. Duda. Mobility Management for Providing QoS in Local Area Wireless Networks. In *DAIS 2001*, 2001. Krakow, Poland.

[53] J.-A. García-Macías, F. Rousseau, G. Berger-Sabbatel, L. Toumi, and A. Duda. Quality of Service and Mobility for the Wireless Internet. In *ACM/IEEE Mobicom 2001, Workshop on Mobile Internet (WMI)*, 2001. Rome, Italy.

[54] R. Guérin and V. Peris. Quality of service in packet networks: basic mechanisms and directions. *Computer Networks*, 31:169–189, 1999.

[55] V. Gupta and S. Glass. Firewall Traversal for Mobile IP.

[56] H. Soliman and C. Castelluccia and K. El-Malki and L. Bellier. Hierarchical MIPv6 mobility management (HMIPv6). `Internet Draft draft-ietf-mobileip-hmipv6-04.txt`, work in progress, july 2001.

[57] J. Haarsten. Bluetooth - the universal radio interface for ad hoc, wireless connectivity. *Ericsson Review*, (3):110–117, 1998.

[58] J. Haarsten. The bluetooth radio system. *IEEE Personal Communications Magazine*, 7(1):28–36, 2000.

[59] R. Handel, N. Huber, and S. Schroder. *ATM Networks: concepts, protocols, applications.* Addison Wesley, 1994.

[60] S. Hanks, T. Li, D. Farinacci, and P. Traina. Generic Routing Encapsulation (GRE). `Internet RFC 1701`, 1994.

[61] J. Heinanen. Protected Best Effort Service. `Internet Draft. draft-heinanen-pbe-svc-01.txt`, 1996.

[62] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. `Internet RFC 2597`, 1999.

[63] A. Helal et al. Towards Integrating Wireless LANs with Wireless WANs using Mobile IP. In *IEEE Wireless Communications and Networks Conference, WCNC*, 2000.

[64] M. A. Hiltzikf. Taming the Wild, Wild Web. Los Angeles Times, july 26, 2001.

[65] C. Huitema. *Routing in the Internet.* Prentice Hall, 1995.

[66] J. Indulska and S. Cook. New RSVP estension to support computer mobility in the Internet. In *Proc. SICON 98*, 1998.

[67] J. Ioannidis. *Protocols for Mobile IP.* PhD thesis, Columbia University, 1993.

[68] D. S. Isenberg. Rise of the Stupid Network. *Computer Telephony*, 4(8), 1997.

[69] V. Jacobson. Congestion Avoidance and Control. *Computer Communication Review*, 18(4):314–329, 1988.

[70] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. `Internet RFC 2598`, 1999.

[71] R. Jain. A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks. *ACM Computer Communications Review*, 19(5):56–71, 1989.

[72] D. Johnson and C. Perkins. Route optimization in mobile ip.

[73] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[74] M. Johnsson. Hiperlan2: The broadband radio transmission technology operating in the 5 ghz frequency band. Hiperlan2 Forum's whitepaper. `http://http://www.hiperlan2.com/web/technology/whitepaper.htm`, 2001.

[75] A. Kamerman. Spread spectrum schemes for microwave-frequency wlans. *Microwave Journal*, 72(4):80–90, 1997.

[76] A. Kamerman and N. Erkocevic. Microwave oven interference on wireless lans operating in the 2.4 ghz ism band. *HF Journal*, (2), 2000.

[77] J. Kempf. Dormant Mode Host Alerting (IP Paging) Problem Statement. `Internet Draft draft-ietf-seamoby-problem-statement-03.txt`, work in progress, may 2001.

[78] J. Kempf. Requirements and Functional Architecture for an IP Host Alerting Protocol. `Internet Draft draft-ietf-seamoby-paging-requirements-01.txt`, work in progress, may 2001.

[79] J. Kempf et al. Dormant Mode Host Alerting ('IP Paging') Problem Statement. `Internet RFC 3132`, 2001.

[80] J. Kempf et al. Requirements and Functional Architecture for an IP Mobile Node Alerting Protocol. `Internet RFC 3154`, 2001.

[81] S. Keshav. *An Engineering Approach to Computer Networking.* Addison Wesley, 1997.

[82] J. Khun-Jush, G. Malmgrem, P. Schramm, and J. Torsner. Hiperlan type 2 for broadband wireless communication. *Ericsson Review*, (2):108–119, 2000.

[83] G. Kirby. Locating the User. *Communications International*, 1995.

[84] L. Kleinrock. *Queuing Systems, Vol. 2: Computer Applications.* Wiley Interscience, 1975.

[85] A. Kopsel, J.-P. Ebert, and A. Wolisz. A performance comparison of point and distributed coordination function of anIEEE 802.11 WLAN in the presence of real-time requirements. In *7th Intl. Workshop on Mobile Multimedia and Communications (MoMuC 2000)*, 2000. Tokio, Japan.

[86] R. Kraut, R. Fish, B. Root, and B. Chalfonte. Reference Model of Open Distributed Processing - Part 3: Prescriptive Model. In *The Claremont Symposium on Applied Social Psychology*, 1990. Intl. Standard 10746-3, ITU-T Recommendation X.903. ITU-ISO. Geneva.

[87] J. Kurose and K. Ross. *Computer networking: A top-down approach featuring the Internet*. Addison Wesley, 2000.

[88] S.-B. Lee, G.-S. Ahn, and A. Campbell. Improving UDP and TCP Performance in Mobile Ad Hoc Networks with INSIGNIA. *IEEE Communication Magazine*, 2001.

[89] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. Campbell. INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks. *Journal of Parallel and Distributed Computing (Academic Press)*, 60(4):374–406, 2000.

[90] S.-B. Lee and A. Campbell. INSIGNIA: In-Band Signaling Support for QOS in Mobile Ad Hoc Networks. In *Proc. 5th Intl. Workshop on Mobile Multimedia Communications (MoMuC), Berlin, Germany*, 1998.

[91] S.-B. Lee, X. Z. G.-S. Ahn, and A. Campbell. Insignia. `Internet Draft draft-ietf-manet-insignia-01.txt`, Work in Progress, November 1999.

[92] S.-B. Lee, X. Zhang, and A. Campbell. Evaluation of the insignia signaling system. In *Proc.8th IFIP Intl. Conf. on High Performance Networking, Paris, France*, 2000.

[93] G. Legrand. *Qualité de Service dans des Environnements Internet Mobile*. PhD thesis, Université Pierre et Marie Curie, Paris VI, july 2001.

[94] G. Legrand, J. Ben-Othman, and E. Horlait. Providing quality of service in mobile environments with MIR. In *Proc. ICON 2000*, 2000. Singapore.

[95] G. Legrand and E. Horlait. An end-to-end QoS architecture for mobile hosts. In *Par. and Dist. Issues in Wireless Nets and Mobile Computing, IFPDPS 01*, 2001. San Francisco, California.

[96] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic. *Proc. SIGCOMM, San Francisco, California*, pages 183–193, 1993.

[97] O. Levkowetz et al. Problem Description: Reason for Performing Context Transfers Between Nodes in an IP Access Network. `Internet Draft draft-ietf-seamoby-context-transfer-problem-stat-01.txt`, work in progress, may 2001.

[98] R.-F. Liao and A. Campbell. On Programmable Universal Mobile Channels in a Cellular Internet. In *4th ACM/IEEE Intl. Conf. on Mobile Computing and Networking*, pages 191–202, 1998.

[99] M. Marsan, C. Chiasserini, R. Lo-Cigno, and M. Munafo. Local and Global Handovers for Mobility Management in Wireless ATM Networks. *IEEE Personal Communications Magazine*, 4(5):16–24, 1997.

[100] D. Mitzel, D.Estrin, S.Shenker, and L.Zhang. An Architectural Comparison of ST-II and RSVP. In *Proc. of IEEE INFOCOM*, 1994.

[101] S. Mukkamalla and B. Raman. Latency and Scaling Issues in Mobile IP, 2001. ICEBERG Project Technical Report, U.C. Berkeley.

[102] A. Myles and D. Skellern. Comparing Four IP Based Mobile Host Protocols. *Computer Networks and ISDN Systems*, 26:349–355, 1995.

[103] J. Nagle. On Packet Switches with Infinite Storage. `Internet RFC 970`, 1985.

[104] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). `Internet RFC 1970`, 1996.

[105] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. `Internet RFC 2474`, 1998.

[106] K. Nichols, V. Jacobson, and L. Zhang. A Two-bit Differentiated Services Architecture. `Internet RFC 2638`, 1999.

[107] C. Nobel. Making 802.11 standards work together. eWeek, july 19, 2000.

[108] T. Noel, D. Grad, J.-J. Pansiot, and S. Marc-Zwecker. Support des communications de mobiles à travers les extensions d'en-têtes d'IPv6. In *Proc. Colloque Francophone sur l'Ingenierie des Protocoles*, pages 45–60, 1997. Liege, Belgique.

[109] A. Orlowski. Microsoft turns the drill on bluetooth. `http://www.theregister.co.uk`, august 01, 2001.

[110] A. Parekh and R. Gallagher. A Generalized Processor Sharing Approach to Flow Control –The Single Node Case. *IEEE/ACM Transactions on Networking*, 1(3), 1993.

[111] A. Parekh and R. Gallagher. A Generalized Processor Sharing Approach to Flow Control –The Multiple Node Case. *IEEE/ACM Transactions on Networking*, 2(2), 1996.

[112] V. Park and S. Corson. Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification. `Internet Draft draft-ietf-manet-tora-spec-04.txt`, Work in Progress, July 2001.

[113] C. Partridge. *Gigabit Networking*, chapter 12. Addison Wesley, 1994.

[114] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[115] D. Pendaraki and R. Guerin. A Framework for Policy-based Admission Control. `Internet RFC 2753`, 2000.

[116] C. Perkins. IP Encapsulation Within IP. `Internet RFC 2003`, 1996.

[117] C. Perkins. Minimal Encapsulation Within IP. `Internet RFC 2004`, 1996.

[118] C. Perkins. Mobile IP Specification. `Internet RFC 2002`, 1996.

[119] C. Perkins. Mobile Networking Through Mobile IP. *IEEE Internet Computing*, 2(1), 1998.

[120] C. Perkins and D. Johnson. Mobility Support in IPv6. In *Mobile Computing and Networking*, pages 27–37, 1996.

[121] C. Perkins and A. Myles. Mobile IP. In *Proc. SBT/IEEE Intl. Telecommunications Symposium, Rio de Janeiro, Brazil*, 1994.

[122] S. Perkins, E. Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. `Internet Draft draft-ietf-manet-aodv-08.txt`, Work in Progress, March 2001.

[123] D. C. Plummer. An Ethernet Address Resolution Protocol. `Internet RFC 826`, 1982.

[124] M. Rahnema. Overview of the GSM System and Protocol Architecture. *IEEE Communications*, 31(4), 1993.

[125] K. Ramakrishnan and S. Floyd. A Proposal to Add Explicit Congestion Notification (ECN) to IP. `Internet RFC 2841`, 1999.

[126] R. Ramjee, T. LaPorta, S. Thuel, K. Varadhan, and L. Salgarelli. IP Micro-mobility Support using HAWAII. `Internet Draft draft-ietf-mobileip-hawaii-01.txt`, work in progress, july 1999.

[127] R. Ramjee, T. LaPorta, S. Thuel, K. Varadhan, and S. Wang. HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks. In *IEEE Intl. Conf. on Network Protocols*, 1999.

[128] J. Roberts. Engineering for Quality of Service. In *Self-Similar Network Traffic and Performance Evaluation*, chapter 16, pages 401–420. Wiley-Interscience, 2000.

[129] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-End Arguments in System Design. *ACM Transactions on Computer Systems*, 2(4), 1984.

[130] J. Schiller. *Mobile Communications*. Addison-Wesley, 1999.

[131] M. Schwartz. Performance Analysis of the SNA Virtual Route Pacing Control. *IEEE Trans. on Communications*, 30(1):172–184, 1982.

[132] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. `Internet RFC 2212`, 1997.

[133] S. Shenker and J. Wroclawski. General characterization parameters for integrated service network elements. `Internet RFC 2215`, 1997.

[134] S. Shenker and J. Wroclawski. Network Element Service Specification Template. `Internet RFC 2216`, 1997.

[135] M. Shreedhar and G. Vargese. Efficient Fair Queueing Using Deficit Round-Robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, 1996.

[136] W. Simpson. IPng Mobility Considerations. `Internet RFC 1688`, 1994.

[137] N. Staff. Psion backtracks on consumer plans. `http://news.cnet.com`, july 12, 2001.

[138] W. Stallings. LAN QoS. *The Internet Protocol Journal*, 4(1):16–23, 2001.

[139] W. Stallings. Mobile IP. *The Internet Protocol Journal*, 4(2):2–14, 2001.

[140] S. G. Steinberg. Netheads vs Bellheads. *Wired Magazine*, 4(10), 1996.

[141] D. Stiliadis and A. Varma. Frame-based Fair Queueing: a new traffic scheduling algorithm for packet-switched networks. In *Proc. Simetrics 96*, 1996.

[142] H. Syed et al. General Requirements for a Context Transfer Framework. `Internet Draft draft-ietf-seamoby-ct-reqs-00.txt`, work in progress, may 2001.

[143] A. Talukdar, B. Badrinath, and A. Acharya. Integrated services packet networks with mobile hosts: architecture and performance. *Journal of Wireless Networks*, 1998.

[144] F. Terqoka, K. Uehara, H. Sunahara, and J. Murai. VIP: A Protocol Providing Host Mobility. *Communications of the ACM*, 37(8):67–75, 1994.

[145] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. `Internet RFC 1971`, 1996.

[146] C. Toh. The Design and Implementation of a Hybrid Handover Protocol for Multimedia Wireless LANs. In *Proc. 1st Intl. Conf. on Mobile Computing and Networking*, 1995.

[147] C.-K. Toh. *Wireless ATM and Ad-hoc Networks*. Kluwer Academic Publishers, 1997.

[148] C. Topolcic. Experimental Internet Stream Protocol, Version 2 (ST-II). `Internet RFC 1190`, 1990.

[149] C. Topolcic. Internet Stream Protocol Version 2 (ST-II) Protocol Specification – Version ST-II+. `Internet RFC 1819`, 1995.

[150] J. Tourrilhes. Linux wavelan drivers. Project website `http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wavelan.html`.

[151] A. Valko. Cellular IP - A New Approach to Internet Host Mobility. *ACM Computer Communication Review*, 1999.

[152] A. Valko, J. Gomez, S. Kim, and A. Campbell. Performance of Cellular IP Access Networks. In *Proc. 6th IFIP Intl. Workshop on Protocols for High Speeds Networks (PfHSN)*, 1999.

[153] N. Valster. Software Interface Specification for Wireless Connection Interface for Wavelan/IEEE (HCF-Light). Doc. no. s0005 rev. 9, Lucent Technologies, january 2000.

[154] N. Valster. Software Interface Specification for Wireless Connection Interface for Wavelan/IEEE (HCF-Light). Doc. no. s0005 rev. 11, Lucent Technologies, july 2000.

[155] W. Weiss. QoS with Differentiated Services. *Bell Labs Technical Journal*, pages 48–62, 1998.

[156] P. White. RSVP and Integrated Services in the Internet: A Tutorial. *IEEE Communications Magazine*, 1997.

[157] W. Willinger, M. Taqqu, and A. Erramilli. *Stochastic Networks*, pages 339–366. Oxford University Press, 1996. A bibliographical guide to self-similar traffic and performance modeling for high-speed data networks.

[158] J. Wroclawski. Specification of the Controlled-Load Network Element Service. `Internet RFC 2211`, 1997.

[159] J. Wroclawski. The Use of RSVP with IETF Integrated Services. `Internet RFC 2210`, 1997.

[160] X. Xiao and L. Ni. Internet QoS: The Big Picture. *IEEE Network*, 13(2):1–13, 1998.

[161] J. Zao et al. A Public Key Secure Mobile IP. In *Proc. ACM Mobicom 97*, pages 173–184, 1997.

Résumé : Cette thèse est dédiée à l'étude de la mobilité et de la qualité de service (QoS) sur les réseaux locaux sans-fil (WLANs). Nous affirmons que plusieurs conditions doivent être satisfaites afin de pouvoir donner des garanties de QoS sur des réseaux locaux sans-fil. Nous proposons que les WLANs adoptent un modèle de QoS basé sur des garanties statistiques, tel que le modèle Diffserv de l'IETF. Le protocole Mobile IP, ainsi que d'autres mécanismes de gestion de la mobilité ne sont pas adaptés pour gérer la mobilité dans des régions limitées géographiquement. Pour cela, nous proposons une approche basée sur le modèle de la micro-mobilité pour assurer la gestion de la mobilité. De même, nous proposons une architecture qui intègre la gestion de la mobilité et de la QoS. Il s'agit d'une architecture hiérarchique qui couple notre protocole de mobilité avec un protocole de signalisation dans la bande (in-band) qui permet la gestion de la QoS. Le mémoire présente d'abord l'état de l'art dans les domaines des WLANs, la mobilité IP, et la QoS sur IP. Ensuite, nous présentons et discutons nos propositions pour la gestion de la QoS et de la mobilité sur les réseaux locaux. De même, nous présentons l'implémentation de prototypes de nos propositions et l'évaluation de leurs performances. Nous concluons par la présentation de nos contributions et des perspectives envisagées.
Mots clefs : réseaux locaux sans-fil, qualité de service, gestion de la QoS, services différenciés, gestion de la mobilité sur IP, micro-mobilité, signalisation in-band.

Abstract: This thesis is devoted to the study of mobility and quality of service (QoS) in wireless local area networks (WLANs). We argue that in order to offer QoS guarantees in WLAN environments, several conditions should be met. We propose that IP-based WLANs should adopt a model based on statistical QoS guarantees, such as the IETF's Diffserv model. Mobility management schemes such as Mobile IP are not adequate for mobility in geographically limited areas and we offer a proposal for mobility management based on a micro-mobility approach. We also propose an integrated architecture for QoS and mobility management. This is a hierarchical architecture that couples our mobility protocol with in-band signaling for QoS management. In this document we present the current state of the art in the areas of WLANs, IP mobility, and IP QoS. Next, our proposals for managing QoS and mobility in local networks are discussed. We then present performance tests and discuss the implementation of our proposals. We conclude by stating our major contributions and foreseen perspectives.
Keywords: wireless LAN, quality of service, QoS management, differentiated services, IP mobility management, micro-mobility, in-band signaling.