ENST

Institut EURECOM

THÈSE

présentée pour obtenir le grade de docteur

de l'école nationale supérieure

des télécommunications

Spécialité : **Informatique et Réseaux**

Neda Nikaein

Mécanismes de contrôle de qualité de service dans des réseaux d'accès IP sans fil

QoS Control in Wireless IP Access Networks

Soutenue le 24 octobre 2000 devant le jury composé de:

Président	Sami Tabbane, SUPCOM
Rapporteur	Philippe Godlewski, ENST
Rapporteur	Michele Zorzi, Universita di Ferrara
Examinateur	Frederic Bauchot, IBM La Gaude
Examinateur	Jouni Mikkonen, Nokia Mobile Phones
Directeur de Thèse	Christian Bonnet

Acknowledgments

First and foremost, I would like to thank my advisor Christian Bonnet for his encouragement and support over the past years. He gave me a lot of freedom with my research as well as the possibility to present my work in several international conferences. I am grateful to the members of my doctoral committee Philippe Godlewski, Michele Zorzi, Frederic Bauchot, Jouni Mikkonen and Sami Tabbane for their time, support and useful comments. My work with Christian Bonnet, Alain Enout and Stéphane Decrauzat on WAND project gave me the chance to gain a valuable experience. I also want to thank all the staff in the Mobile Communications Department and Institut Eurécom for their warm reception.

There is more to life than just work. I was fortunate to have a lot of friends who are of great value for me. My special thanks go to Zoe O'Brien for her true friendship and almost 3 years of "common life". I also want to thank Perrine Delacourt for her helpful advices and for her presence on my defense day, Franck Thely for his lessons on snow boarding and the wonderful winter holidays, Hubert Gueguen for his unexpected tie on my defense day, Irfan Ghauri for learning me how to be a real cowboy, Frederique Olivier and Valerie Galassi for our amusing gossip dinners, Pierre Abel and Sophie Larrouy for their usual "Baptême" parties, Stéphane Decrauzat for his daily cheerful jokes, swiss cheese and "Fondue", Alain Enout for his lessons on "argot" French slang language, Giuseppe Montalbano and Maria-Luisa Parlatano for their glamorous wedding, Daniela Tuninetti for supporting me in the office and her helpful comments on my presentation, Houda Labiod for reading this thesis and her comments on my work, and all the PhD students at Eurécom that I did mention.

I owe a special gratitude to my family. My mother Parvin Alizadeh was a great source of motivation and encouragement for my studies. I owe her all that I have accomplished up to now and I want to dedicate this thesis to her. I want to thank my uncle Ali Alizadeh for helping me with settling down in France and his support during my hard student life in Paris. I am indebted to my brother and friend Navid Nikaein for reading this thesis and his help with my "List of Acronyms" and his support during my hard days.

Finally, my "koochooloo" Sergio Loureiro gave me his infinite support and love. I am grateful to him for enriching my life. I only hope to play the same role in his life.

Résumé

Les technologies de l'information ont évolué énormément pendant ces vingt dernières années. Le progrès a été réalisé dans deux axes principaux: systèmes sans fil et l'Internet. Les systèmes sans fil sont devenus populaires en raison des services mobiles qu'ils offrent à leurs utilisateurs. L'Internet, d'autre part, doit sa popularité au fait qu'il offre une interface commune aux protocoles de couches hautes indépendemment du type de liaison. En conséquence, il est raisonnable d'étudier les conséquences de la combinaison de ces deux technologies afin de permettre aux utilisateurs d'avoir accès à l'Internet dans des environnements sans fil. Les systèmes cellulaires actuels ne peuvent pas satisfaire les demandes de Qualité de Service (QoS) des applications multimédia dues à leurs bas débits. C'est pourquoi nous nous situons dans le cadre de systèmes d'accès sans fil large bande.

Dans cette thèse, nous examinons d'abord comment nous pouvons apporter le trafic IP dans des réseaux sans fil. Nous présentons une description complète d'un réseau d'accès sans fil apportant des applications IP aux terminaux mobiles. Le système fournit un accès sans fil IP avec un débit élevé et une mobilité limitée. Nous nous concentrons sur trois problèmes notamment le support de QoS, le support de mobilité et le support de multicast dans ce réseau d'accès.

Nous proposons une architecture pour le support de QoS dans le réseau d'accès. Dans cette architecture, le mécanisme de QoS du réseau d'accès est complètement indépendant du mécanisme de QoS du réseau principal. Dans le réseau d'accès, nous avons défini trois classes de QoS au niveau de lien radio. Le trafic entrant est classifié dans ces trois classes de QoS dans le réseau d'accès. La différenciation de service dans le réseau d'accès est faite en utilisant différentes files d'attente et différents mécanismes de fiabilité en fonction de QoS désirée.

Nous présentons un cadre pour la gestion de mobilité dans le réseau d'accès. Le *handover* au sein du même sous-réseau IP est géré par la couche liaison radio tandis que le *handover* entre différents sous-réseau IP est basé sur le mécanisme de gestion de la mobilité IPv6. Le *handover* au sein du même sous-réseau IP peut être effectué assez rapidement puisque toutes les caractéristiques du trafic du terminal mobile sont déjà disponible au routeur. Le *handover* entre différents sous-réseaux IP est plus compliqué puisque le terminal mobile change de domaine IP et obtient par conséquent une nouvelle adresse. Dans ce cas, le nouveau routeur doit être informé des caractéristiques des trafics du mobile avant de les transmettre sur son réseau local. La gestion de mobilité offert par IPv6 est adapté à la mobilité du type *macro* où le mobile ne change pas son point de connexion à l'Internet fréquemment. En outre, ces *handovers* peuvent entamer des pertes de paquets pendent le temps où le nouveau routeur n'est pas encore informé de la nouvelle adresse du mobile. Afin de réduire ces pertes, le

routeur local du réseau dans lequel le mobile était situé avant le *handover* peut intercepter le trafic destiné au mobile pour le renvoyer ensuite vers le nouvel emplacement du terminal mobile.

Une autre question importante en cas de *handover* entre différents sous-réseau IP est le problème de support de QoS. Nous avons identifié un problème d'adressage entre le mécanisme de gestion de mobilité en IPv6 et le mécanisme de support de QoS. Ce problème surgit seulement si le mécanisme du support de QoS est basé sur l'architecture *Integrated Services*. L'IPv6 propose l'utilisation d'une adresse temporaire pour la localisation d'un terminal mobile en dehors de son réseau local. Le protocole de signalisation (RSVP) utilisé dans l'architecture *Integrated Services* ignore la mobilité et le changement de l'adresse du mobile. En conséquence, nous observons un problème de routage ainsi qu'une dégradation de QoS chaque fois qu'un mobile change son adresse IP. Nous avons proposé une solution complète pour résoudre ces problèmes. En effet, en raison du changement de l'adresse du mobile, les réservations faites dans les routeurs intermédiaires tout au le long de la route entre le mobile et le noeud avec lequel il communique doivent être mises à jour afin de refléter la nouvelle adresse du mobile.

Nous présentons par la suite un cadre complet pour la transmission multicast dans le réseau d'accès. Nous avons proposé un nouveau protocole de gestion de groupe afin de tracer les membres de groupe dans le réseau d'accès. Les résultats numériques ont prouvé que ce protocole est plus efficace que l'IGMP en termes d'utilisation de bande passante. Nous avons également proposé de transmettre le trafic multicast seulement aux Points d'Accès (AP) qui gèrent les membres actifs du groupe auquel le trafic est destiné. Le cadre proposé est complété par un système d'adressage au niveau de la couche liaison. Cette adressage permet d'identifier des groupes dans chaque AP. L'AP alloue une adresse à un groupe s' il a au moins un membre du groupe dans sa cellule. Il communique également cette adresse aux membres de groupe dans sa zone de couverture. Cette adresse est unique dans chaque AP.

Les liens sans fil souffrent d'un taux d'erreurs élevé et d'une faible capacité de bande passante. En conséquence, les mécanismes de support de QoS ne fonctionneront pas efficacement à moins qu'ils ne soient couplés avec des protocoles de contrôle d'erreurs adaptés. Nous étudions l'utilisation de différents protocoles de contrôle d'erreurs et différents paramètres de codage afin de contrôler le QoS au réseau d'accès. Nous présentons plusieurs résultats numériques en termes de plusieurs paramètres de QoS pour différents mécanismes de contrôle d'erreurs. Ces résultats numériques prouvent que l'utilisation de FEC peut améliorer la performance du mécanisme de contrôle d'erreurs dans la plupart des cas. Cependant choisir un code qui a une performance optimale dans tous les états de canal et pour tout nombre de récepteurs sans fil est une tâche difficile. Par conséquent, un protocole efficace de contrôle d'erreurs doit pouvoir changer ses paramètres de codage ainsi que sa stratégie de contrôle d'erreurs d'une façon dynamique.

Nous proposons un mécanisme adaptatif de contrôle d'erreurs basé sur la notion de QoS et l'estimation de l'état du canal radio. Le protocole prévoit l'évolution des conditions de canal de chaque récepteur. Basé sur cette prévision et les besoins de QoS des récepteurs, il change sa stratégie de contrôle d'erreurs aussi bien que ses paramètres de codage. Les résultats de simulation comparent la performance de notre protocole adaptatif de contrôle d'erreurs aux protocoles fixes. En général, nous observons que notre protocole adaptatif réduit l'utilisation de la bande passante tout en respectant les besoins de QoS des récepteurs.

Abstract

The Information technology has evolved enormously over the last twenty years. Progress has been made in two major axes: wireless systems and Internet. Wireless systems have become popular due to their offer of ubiquitous services to end users. Internet, on the other hand, owes its popularity to the fact that it offers a common interface to higher layer protocols over a wide range of communication links. Therefore, it is interesting to investigate the implications of combining these two technologies in order to enable users to have access to Internet in wireless environments. Current cellular systems can not meet the QoS demands of multimedia applications due to their low data rates. New wireless broadband network techniques must be developed in order to meet these demands.

In this dissertation, we first examine how we can bring IP traffic in wireless networks. We present a complete description of a wireless IP access network bringing IP applications to mobile terminals. The system provides a short range high data rate wireless access to IP. We focus on three issues namely QoS support, mobility support and multicast support in the access network. We provide a framework as well as a mechanism for the access network to support each of these design objectives.

Wireless links suffer from high error rate and low bandwidth capacity. As a result, the QoS support mechanisms will not work efficiently unless coupled with appropriate error control protocols. We investigate the use of different error control protocols and different coding parameters in order to control QoS at the access network. We present several numerical results in terms of several QoS metrics for different error control mechanisms. These numerical results show that the use of FEC can improve the performance of the error control mechanism in most cases but choosing a code that can perform efficiently in all channel conditions and for any number of wireless receivers is a difficult task. Therefore, an efficient error control protocol must be able to change its coding parameters dynamically.

We propose an adaptive QoS-based error control mechanism. The protocol predicts the evolution of the channel conditions of each receiver. Based on this prediction and the QoS requirements of the receivers, it changes its error control strategy as well as its coding parameters. Simulation results compare the performance of our proposed adaptive error control protocol with fixed error control protocols. In general, we observe that our adaptive protocol reduces the bandwidth utilization while respecting the QoS requirements of the receivers.

Contents

List of Figures

List of Tables

List of Acronyms

AAL5	ATM Adaptation Layer
ACTS	Advanced Communications Technology and Services
AF	Assured Forwarding
AP	Access Point
APCP	Access Point Control Protocol
ARQ	Automatic Repeat reQuest
ATM	Asynchronous Transfer Mode
BE	Best Effort
BPSK	Binary Phase Shift Keying
BRAN	Broadband Radio Access Network
BSC	Binary Symmetric Channel
CAC	Connection Admission Control
CBT	Core Based Tree
CL	Controlled Load
CLIP	Classical IP
CN	Correspondent Node
CPN	Customer Premises Networks
CRC	Cyclic Redundancy Check
coa	care-of-address
CSCW	Computer Supported Collaborative Work
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services architecture
DS	Differentiated Services
DSP	Digital Signal Processor
DVMRP	Distance Vector Multicast Routing Protocol
EDF	Earliest Deadline First
EF	Expedited Forwarding
ETSI	European Telecommunication Standard Institute
FEC	Forward Error Correction
FC	Flow Compression
FCAC	Fixed network Connection Admission Control
FIFO	First In First Out
filter spec	filter specification
FTP	File Transfer Protocol

GE	Gilbert-Elliot
GLI	Group Location Information
GMI	Group Membership Information
GRAN	General Radio Access Network
GS	Guaranteed Service
ha	home address
HA	Home Agent
HIPERLAN	HIgh PErformance Radio Local Area Network
IC	Integrated Circuit
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IFMP	Ipsilon Flow Management Protocol
IntServ	Integrated Services architecture
IP	Internet Protocol
ISP	Internet Service Provider
IWF	Inter-Working Functionality
LAN	Local Area Network
LANE	Local Area Network Emulation
MAC	Medium Access Control
MCA	MultiCast Agents
MCP	Mobile Control Protocol
MRSVP	Mobile Resource ReServation Protocol
MMC	Mobility Management Controller
MOSPF	Multicast Open Shortest Path First
MPLS	MultiProtocol Label Switching
MT	Mobile Terminal
MR	Mobility Enhanced IP Router
OPWA	One Pass With Advertising
PDU	Protocol Data Unit
PHB	Per-Hop Behavior
QoS	Quality of Service
RAN	Radio Access Network
RRM	Radio Resource Manager
RS	Reed-Solomon
RSE	Reed-Solomon Erasure
RSVP	Resource ReServation Protocol
RTD	Round Trip Delay
SAR	Segmentation And Reassembly
SNR	Signal to Noise Ratio
SRCM	Stochastic Radio Channel Model
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TDD	Time Division Duplex
UDP	User Datagram Protocol
VPI/VCI	Virtual Path Identifiers/ Virtual Circuit Identifier

UMTS	Universal Mobile Telecommunications System
WAND	Wireless ATM Network Demonstrator
WCAC	Wireless network Connection Admission Control
WFMP	Wireless Flow Management Protocol
WGMP	Wireless Group Management Protocol
WLAN	Wireless Local Area Networks
WWW	World Wide Web

Chapter 1

Introduction

The Information technology has evolved enormously over the last twenty years. At one side, we see the explosive growth of Internet and at the other side, we see the astonishing success of wireless communication systems. The success of *Internet Protocol* (IP) is due to the fact that it offers a common interface to higher layer protocols over a wide range of communication links while the popularity of wireless systems is due to their offer of ubiquitous services to end users. The most popular wireless communication system is *cellular telephony* which offers a high level of mobility with a limited data rate. The other wireless system that is also evolving rapidly is *Wireless Local Area Networks* (WLAN). This technology offers a high data rate at the expense of a limited mobility.

Recently, the emergence of new multimedia applications has created a strong need for broadband networks with high data rates and support for *Quality of Service* (QoS). In response, the Internet is moving from a best effort model to a system, capable of supporting a range of traffic characteristics and service requirements. The mobility features of IP has also increased its suitability for wireless systems. Therefore, it is unavoidable to think about the implications of combining these two technologies in order to enable users to have access to Internet in wireless environments. However, two main obstacles exist. First of all, Internet traffic has long been carried out by wired links having a low error rate and an ever increasing bandwidth. Wireless links however suffer from high error rate and low bandwidth capacity. Most of the protocols designed for IP assume an error free medium. These protocols show a high performance degradation when used on wireless links. Secondly, current wireless networks do not provide sufficient QoS support for multimedia applications. To meet this increasing user demands, new wireless broadband network techniques have to be researched and developed.

In this dissertation, we first examine the issue of how we can bring IP traffic in wireless networks. The result of this work is a complete description of a wireless IP access system bringing IP applications to mobile terminals. Due to the enhanced features offered by IPv6 such as integrated mobility management with optimized routing and address auto-configuration mechanism, the developed system architecture is based on IPv6 rather than IPv4. This part has been carried out within the *Wireless ATM Network Demonstrator* (WAND) project. In the second part, we try to improve and propose some new solutions and mechanisms to control QoS in the proposed wireless IP access system.

This chapter serves as a general introduction to the whole dissertation. We start by presenting the problems that we studied as well as our research contributions. Then, we provide the organization of the remainder of the dissertation.

1.1 Scope and Contributions

Our research presented in this dissertation is composed of two complementary parts. The first part consists of the design of a wireless IP access system which is carried out within the WAND project. The main problems studied and our contributions in this part are:

- **What are the main entities of a wireless IP access system and what are their functionalities?**
  The system architecture follows the outlines of the theoretical *General Radio Access Network* (GRAN) reference model defined in the *Universal Mobile Telecommunications System* (UMTS). The main entities of the system are *Mobile Terminal*, *Access Point* and *Mobility Enhanced IP Router*. The mobile terminal is the end point of the Internet and all radio access network control protocols. The access point is in charge of all the radio dependent control functionality and the mobility enhanced router is the boundary of the wireless IP system providing access to the Internet backbone. Each entity includes several functional blocks interfacing to other blocks. These issues are discussed in Chapter 2. Our contributions are the participation to the overall architecture design of the system, the design of the multicast functional block as well as the description of its interfaces with other entities.

- **How can we support the IP QoS mechanisms at the access network?**
  The main problem of QoS support in the access network is the lack of a homogeneous standard for IP QoS. Therefore, we study the main IP QoS mechanisms. For each mechanism, we propose a solution to support QoS at the access network. The whole approach is then based on the separation of the radio access QoS scheme from the IP backbone QoS technique. In fact, we only provide a flexible mechanism capable of mapping the fixed network QoS parameters into the radio access QoS classes. At the radio access network, we define three different QoS classes corresponding to different delay constraints. Due to the high bit error rate of the wireless links, most of the wireless systems are equipped with a complementary error control protocol at the link layer. The error control mechanism must be chosen as a function of QoS requirements of a packet. A packet having a strict delay constraint, does not need to be retransmitted after its deadline. Therefore at the link layer, we propose three different reliability mechanisms for each radio QoS class. The problem of QoS support is studied in Chapter 3. Our contributions are the participation at the early specification of the QoS architecture and its general framework.

- **How can we support mobility inside and outside the access network and what is the effect of mobility on QoS?**
  Mobility is an inherent part of the system. Mobility within the access network is supported by the radio sub-system while mobility between different access networks follows the mobility management mechanisms of IPv6. We identify an address mismatch

problem when using IP QoS mechanisms in the case of an inter-subnet handover. In fact, IP proposes the use of temporary IP addresses in order to locate a mobile terminal outside its home network. Some IP QoS mechanisms however are not aware of the address change of a mobile terminal causing routing problems and QoS degradation each time that a mobile terminal acquires a new temporary address. We propose a solution with some optional enhancements in order to overcome the routing problem and the QoS degradation in these situations. The problem of mobility and its effect on QoS is discussed in Chapter 4. Our contributions are the participation at the problem identification and the proposal of solutions to resolve the problem of IP QoS protocol mismatch with IP mobility features in case of multicast traffic.

- **How can we support multicast applications at the access network and what is the effect of mobility on multicast communication?**
  Due to the popularity of multicast applications, multicast support is one of the design goals of the wireless IP access system. In order to avoid sending a separate copy of a packet to each receiver of a group, we take advantage of the broadcast nature of the radio medium to transmit the packet. At the radio sub-system, a link layer addressing scheme is used for multicast traffic. We also study the problems of IP multicast in the presence of wireless links. In order to resolve these problems, we propose a new group management protocol. We provide some performance evaluations and we show that our proposed protocol has less overhead than its counterpart in the IP multicast standard. Some solutions are also proposed in order to avoid routing problems when a mobile terminal participating in a multicast session moves to another network. These issues are investigated in Chapter 5. Our contributions are the problem identification and the proposal of a new group management protocol, the performance evaluation of the protocol and its counterparts in the IP multicast standard and the proposal of solutions for multicast communications in case of terminal mobility.

The second part of this dissertation consists of the use of adaptive coding for QoS control. Our research in this part is carried out considering a multicast communication mode where data is sent to a set of receivers. Other communication modes such as unicast and broadcast can be viewed as special cases where data is sent to one or all receivers. Having a framework for QoS control of multicast communication means that the same general basis can be applied to any communication mode. This part has been done after the completion of the WAND project. The main problems studied and our contributions in this part are:

- **What is the effect of coding on QoS and how can we choose the appropriate parameters for a coding scheme as a function of the QoS requirements of a packet?**
  We focus on Reed-Solomon Erasure codes and we take three different QoS metrics in order to evaluate the performance of coding. These QoS metrics are bandwidth, average delay of a packet and the loss rate. Each QoS metric is evaluated for different coding parameters. We take an independent error model and a bursty error model. The numerical results in each case show that we can not choose a single best code for all situations. Depending on QoS requirements, the channel bit error rate and the number of receivers, the choice of best code varies and therefore, it may be useful to investigate adaptive coding. The proposed adaptive coding scheme must be able to change its parameters as a function of channel bit error rate, QoS and number of receivers. This

problem is studied in Chapter 6. Our contributions are the performance evaluation of different Reed-Solomon Erasure codes in terms of bandwidth use, delay and loss rate in the presence of independent and bursty error models.

- **How well can an adaptive coding scheme help us to control the QoS at the radio channel?**
  Based on the numerical results of Chapter 6, we propose a QoS-based adaptive coding scheme. The adaptive algorithm changes the coding parameters as a function of the desired QoS, channel bit error rate and the number of receivers. Simulations show that our proposed adaptive scheme provides better results than other fixed schemes. The adaptive algorithm as well as the simulation results are presented in Chapter 7. Our contributions are the proposal of an adaptive coding scheme and the simulations comparing the performance of our proposed adaptive mechanism with other schemes.

1.2 Structure of Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 presents the architecture of the wireless IP access system developed in the framework of the WAND project. After fixing the design objectives and studying the implications of having an IP based access system, we provide the general system architecture. We then present a functional description of the whole system. The details of how each functionality is supported by the system is treated in the next chapters.

Chapter 3 discusses the problem of QoS support in the system. We start by presenting the two main QoS approaches used in Internet. We then present the concept of IP flow which enables us to classify the packets requiring a certain QoS at the radio access network. After defining our radio link QoS classes, we propose a mapping method in order to map the IP QoS parameters into the radio QoS classes.

Chapter 4 provides the mobility management scheme used in the system. We give a brief overview of IPv6 mobility management scheme. Then we explain the basics of inter-subnet and intra-subnet handover schemes. We tackle the problem of routing and QoS degradation during handover specially during an inter-subnet handover. We identify an address mismatch problem between the IP QoS approaches and IPv6 mobility scheme and we provide a solution for this problem.

Chapter 5 presents the multicasting scheme of the wireless IP access system. We first describe the IP multicast standard and the specific characteristics of the wireless links requiring some modifications in the IP multicast standard. Then, we propose a new group management protocol. We provide numerical results and we show that our proposed scheme is more suitable in wireless environments due to its lower overhead. We also study the effect of mobility on multicast communication.

Chapter 6 investigates the effect of coding on QoS control. We start by some background information on coding and Reed-Solomon Erasure codes. Then we explain the QoS metrics that we use to evaluate the performance of each code. We take two different error models in our performance evaluations: an independent error model and a bursty error model.

We present some numerical results for both error models for two scenarios. The first scenario corresponds to a pure retransmission scheme while the second corresponds to a hybrid mechanism combining coding with retransmission.

Chapter 7 presents our adaptive coding algorithm. We first explain the finite state Markov model used throughout this chapter to simulate the radio channel. Then we present our adaptation algorithm and some simulation results comparing its performance with other schemes.

Finally Chapter 8 summarizes our results and our contributions. We conclude with directions for further research.

Chapter 2

WAND Wireless IP Architecture

The WAND project was originally designed to provide a 20 Mbit/s, 5.2 GHz radio access to *Asynchronous Transfer Mode* (ATM) backbone as it was globally expected that ATM will become the de-facto networking technology even in *Local Area Networks* (LAN). In the WAND reference design, IP services are provided using *LAN Emulation* (LANE) [ATM97] [Jef94] which separates IP packets with different QoS requirements but handles all IP traffic as best effort data [MAA+98]. *CLassical IP* (CLIP) technology [LH98], where IP packets are directly encapsulated into *ATM Adaptation Layer* (AAL5) frames was not deployed as it will not provide any QoS mechanisms. Furthermore, neither CLIP nor LANE support delay sensitive multimedia.

At the moment, the Internet has started changing from a best effort services model to an integrated services model with a wide range of applications and different traffic characteristics. The introduction of new innovative QoS aware IP techniques has significantly improved the usability of multimedia and real-time applications in the Internet. The integrated mobility features of IPv6 [PJ96] and mobile IP [Per98] increase the suitability and attractiveness of the IP protocol even more in mobile networks. Many experts foresee that instead of ATM, IP will maintain its position as the leading access network technology in fixed LANs. Consequently, the future wireless networks should offer native IP access over high speed indoor radio link and reliably maintain IP QoS characteristics over the air interface.

In order to provide an efficient wireless access to the Internet backbone, the WAND project started to specify a new mechanism which is optimized for transmitting native IP traffic over the 5.2 GHz radio link. The target was to minimize the IP protocol overhead in the wireless interface and to seamlessly support IPv6 and the state-of-the-art IPv6 QoS mechanisms. The proposed concept enables full exploitation of real-time IP applications in mobile environment thus increasing the applicability of the developed 5.2 GHz WAND radio sub-system.

IPv6 was seen as a relevant starting point for the future wireless broadband system as its new features will provide integrated support for terminal mobility. These features include built-in mobility with optimized routing, address auto-configuration and IP security mechanisms. These features enhanced with appropriate wireless access network specific modifications will enable IP based mobility between IP subnets, secure communications and provide registration support mechanisms for wireless IP networking. Due to the enhanced

features offered by IPv6, the developed system architecture will be based on IPv6 rather than IPv4.

This chapter presents the architecture of the wireless IP system designed in the framework of the WAND project. We start by presenting the WAND project and its original environment. We then provide the motivations behind the design of a wireless IP system. After highlighting the design objectives of the wireless IP system, we present a general overview of the main entities of the system as well as their functionalities. Finally, we discuss the system level as well as the functional level design issues.

2.1 WAND Environment

Advanced Communications Technology and Services, known simply as ACTS, is one of the specific programs of the *Fourth Framework Program* of the European Community activities in the field of research and technological development and demonstration (1994 to 1998). In fact, the European Union's research effort is focused on accelerating the deployment of advanced communications infrastructures and services. This effort is complemented by extensive European research in the related fields of information technology and telematics. The Magic WAND project AC085 was one of the ACTS projects. It was a three-year research project which ended in September 1998. The project was composed of industrial and academic partners listed below. Project administrative management as well as technical management were assumed by Nokia Mobile Phones.

Partner	**Status**	**Country**
Nokia Mobile Phones	Industrial	Finland
VTT Information Technology	Industrial	
Tampere University of Technology	Academic	
IBM France	Industrial	France
Institut Eurécom	Academic	
Robert Bosch	Industrial	Germany
University of Ulm	Academic	
Intracom	Industrial	Greece
University of Athens	Academic	
Lucent Technologies WCND	Industrial	Netherlands
Ascom Tech	Industrial	Switzerland
ETH Zürich	Academic	

ATM is a cell-based switching and multiplexing technology designed to be a general-purpose connection-oriented transfer mode for a wide range of services. The short length of the cells allows rapid switching of any mixture of traffic such as audio, video and bursty data traffic at varying bit rates. The connection oriented nature of ATM allows the user to specify certain QoS parameters such as loss rate and delay for each connection.

Given that ATM was supposed to be the future backbone technology, several research projects started investigating on wireless ATM. Wireless ATM is mainly considered as an

access to an ATM network. WAND goals were to specify, develop and implement a wireless access system for ATM that extends the service characteristics and benefits of the ATM technology to mobile users. Communication between the mobile, portable terminals and the access points takes place in the 5.2 GHz frequency range at a nominal transmission speed of 20 Mbit/s and a range of up to 50 meters.

The three main objectives of the project were:

- to specify a wireless customer premises access system for ATM networks,
- to promote the standardization of wireless ATM access systems,
- and to demonstrate and carry out user trials with the selected user group in order to test the feasibility of a radio based ATM access system.

In terms of user trials, the feasibility of a wireless ATM access system was verified in hospitals and office environments. The project also actively promoted the standardization process of in *European Telecommunication Standard Institute* (ETSI). The positioning of WAND compared to other mobile systems is depicted in Figure 2.1.

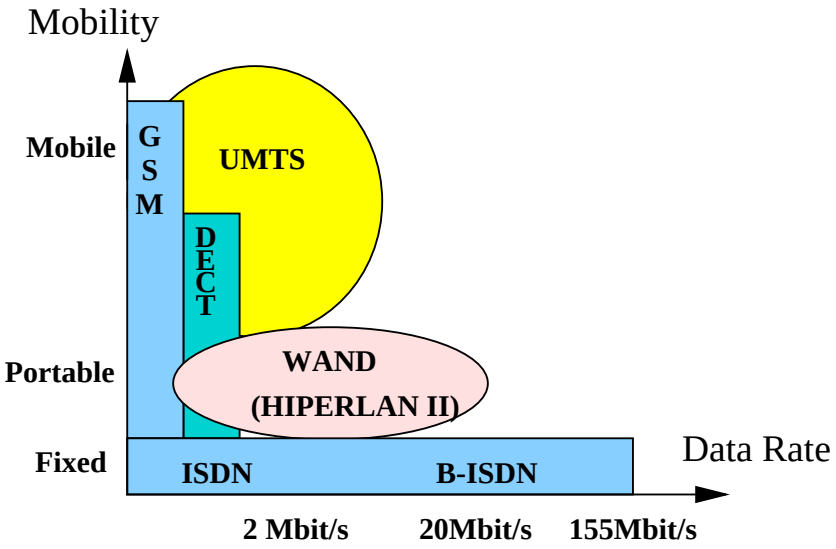


Figure 2.1: Positioning of WAND

2.2 WAND and the Standardization Process

The increasing demand for multimedia applications created a need for broadband technology for wireless networks. The range of data rates actually supported by digital cellular systems can not meet the bandwidth demands of true high resolution multimedia applications. Furthermore, it is not clear that there exists demand for such services beyond the premises of a business or other organization. In response to these demands, ETSI created the *Broadband Radio Access Network* (BRAN) group to specify new standards for wireless broadband systems.

Two major standards specified by BRAN are HIPERLAN I [BRA98] and HIPERLAN II [BRA00] systems. HIPERLAN stands for HIgh PErformance Radio Local Area Network. HIPERLAN I provides a high speed radio local area network which is compatible with *Ethernet* LAN. The frames have the same headers and the frame size is up to 1500 bytes. However, the *Medium Access Control* (MAC) protocol is based on *Carrier Sense Multiple Access with Collision Avoidance* CSMA/CA instead of CSMA/CD of the Ethernet LAN since *Collision Detection* (CD) is quite expensive in terms of bandwidth for radio channels. The user mobility is limited within the local area network. HIPERLAN I provides a connectionless best effort frame delivery. It is also compatible with IEEE 802.11 [IEE99] which is the American standard for wireless LANs.

HIPERLAN II provides short range broadband wireless access to IP, ATM, and UMTS. It can support multimedia applications due to its ability to provide QoS including required data rates that a user expects from a wired IP or ATM network. Its typical operating environment is indoor but wide area mobility may also be supported by other standards outside the scope of BRAN group. The MAC protocol of HIPERLAN II is based on a *Time Division Multiple Access/Time Division Duplex* TDMA/TDD scheme with centralized control by a scheduler determining the frame composition. The MAC frame appears with a period of 2 msec. The main *Protocol Data Unit* (PDU) of the system has a fixed size of 54 bytes. WAND is one of the research projects that participated actively in the standardization process of the HIPERLAN II in BRAN group. HIPERLAN II has inherited its access method (TDMA/TDD) as well as its packet size of 54 bytes from WAND.

2.3 Design Objectives

The developed wireless IP access system is targeted for *Customer Premises Networks* (CPN) and *Business LANs*. Here the WAND system provides a wireless extension to the existing fixed LAN infrastructure. As a result, the major design criteria of the system are:

- **Quality of Service Support**

  The developed wireless IP system must be capable of delivering multimedia data streams while respecting their QoS requirements. In this sense, it has to provide a mechanism to detect and differentiate various traffic flows such as time-critical data flows and best effort data. The system should be compatible with the state-of-the-art fixed network IP QoS mechanisms.

- **Mobility Management**

  The wireless network must support mobility both within the IP subnetwork (intra-subnet mobility) and between different IP subnetworks (inter-subnet mobility). The mobility management scheme covers both radio access network and core network. The radio access network mobility management must be based on the handover scheme specified in the original WAND design while the core network mobility must use the standard integrated IPv6 mobility features.

- **Multicast Support**

Multicast applications such as video conferencing and multi-player games are becoming increasingly popular in the Internet. Hence, our wireless IP system must support multicast communication. This translates into a need for a multicast group management mechanism and a way to transmit traffic to multiple senders. The system also needs to cooperate with Internet multicast routing protocols in order to deliver multicast traffic to other networks.

- **Compatibility**

  The proposed solution must maximize scalability and compatibility with the developed WAND radio. The radio access network can be connected to any standard fixed IP core network. The QoS management functions are defined as extensions to the standard IPv6 protocol stack and the network mobility scheme follows the IPv6 mobility scheme.

2.4 Theoretical Reference Model

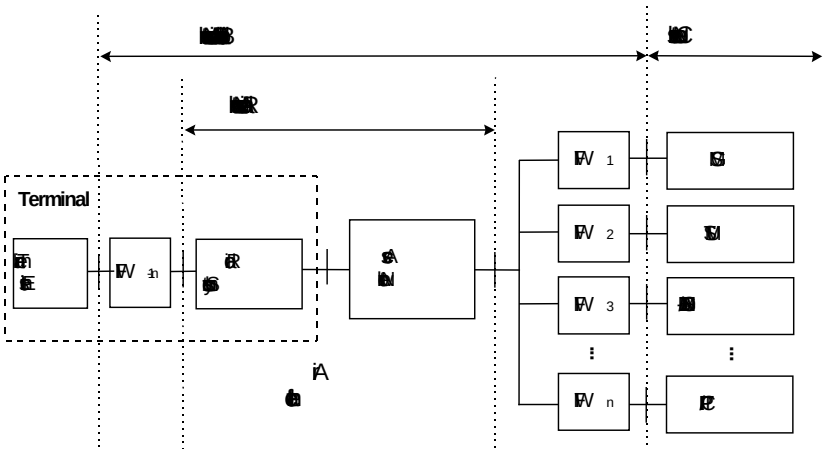


Figure 2.2: General GRAN reference model [SALM+98]

The system architecture follows the outlines of the theoretical GRAN reference model, illustrated in Figure 2.2. The GRAN broadband radio access network reference model, defined in the UMTS concept, consists of a *Radio Access Network* (RAN), a *Broadband Radio Access Network* (BRAN) and different core networks. The BRAN includes the RAN and core network dependent *Inter-Working Functionality* (IWF) blocks. The RAN covers all of the radio dependent parts and the IWFs link the RAN to various core networks and terminal entities. Hereafter the term wireless IP system refers to the entire BRAN network covering RAN and necessary IWFs. The wireless IP system is connected to the IP core network.

2.5 Network Entities

The network, depicted in Figure 2.3, is composed of *Mobile Terminals* (MT), *Access Points* (AP), and *Mobility enhanced IP router* (M-router or MR):

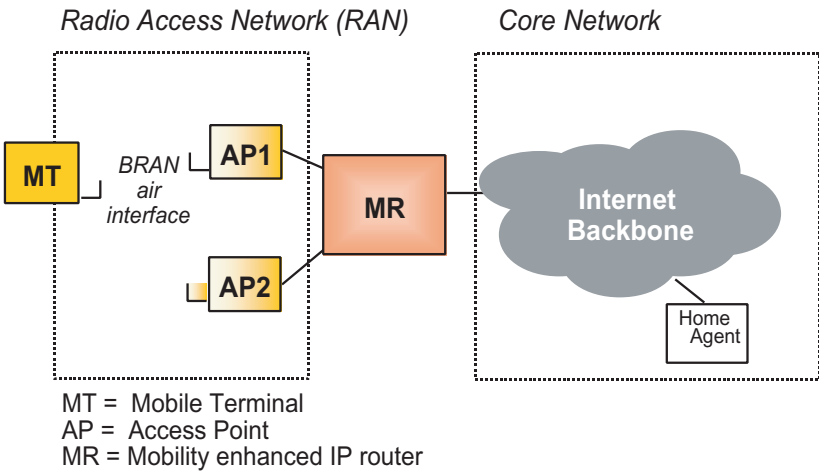


Figure 2.3: General system architecture [SALM+98]

- *Mobile Terminal* (MT) is the user device for accessing wireless Internet services. The terminal will be the end point of the Internet and radio access network control protocols.
- *Access Point* (AP) implements all the radio dependent control functionality. APs include radio resource management and radio link control functions.
- *Mobility enhanced IP router* (MR) is the boundary of the wireless IP subnetwork and manages one or more APs. It handles the mobility and location management of terminals that are registered to APs. It also provides IP mobility services and address allocation functions.
- *Radio Access Network* (RAN) implements all the radio dependent functionality such as radio resource management, set-up and release of wireless flows and handovers. RAN contains MTs and APs.
- *Home Agent (HA)* - The IP home agent functionality is needed to provide mobility between wireless IP subnets. Note that the functions of the home agent are as in standard Mobile IP. The HA resides in the home network of the MT and is accessed through standard IP gateways. Typically, the HA is implemented as part of the MR of the home network. However it can also be a separate entity. Potentially the HA could be extended to contain user authentication information and a billing database (assuming a one-to-one association between user and MT).

Reference points for interoperability are assumed to exist in three places:

- Radio link interface (MT - AP)
- Core network interface (AP - MR)
- Terminal - core network interface (MT - MR)

The radio interface is assumed to follow the BRAN HIPERLAN type II standardization. The standardization of the AP-MR interface will allow users to purchase their radio access network from any manufacturer. The MT-MR interface is a logical interface. Its standardization will allow the same MR device to be accessed by different types of terminals.

2.6 System Architecture

Figure 2.4 depicts a general overview of the system architecture. The MR has full TCP/IP protocol functionality. It performs standard IP routing forwarding packets to the RAN interface and embeds wireless specific control functions. The MR classifies incoming IP packet flows and relays them via the corresponding AP to the MT using suitable QoS characteristics. It also controls IP flows, terminal mobility and location management. The MR controls the APs using a specific control protocol. The AP implements a LAN bridge multiplexing different IP traffic on the radio channel.

The MT includes all standard TCP/IP protocols and wireless specific control services. The control messages are transparently sent between the MR and terminals using control functions.

The MR segments and reassembles IP packets into segments that fit into radio link packets. The *Segmentation And Reassembly* (SAR) blocks handle the segmentation of packets between the MT and the MR. The AP only transparently relays the segmented packets between the radio access network and the fixed network. The system utilizes ATM like segmentation, which segments the IP packets into 48 bytes cells. Normally an ATM cell has a payload of 48 bytes and a header of 5 bytes. At the link layer, the ATM header is increased to 6 bytes giving a 54 bytes data packet at the radio level. This extra byte is used to carry error control information.

Given that the current WAND system is ATM based, one of the main design objective is to provide wireless broadband IP access with minimal modifications to the existing WAND radio. Therefore, in the data plane, ATM cells were maintained since the small fixed size cell nature of ATM works well in terms of efficient multiplexing and forwarding speeds. However, ATM connection-oriented control protocols are not appropriate for native IP traffic. Therefore, ATM control is replaced with the *Wireless Flow Management Protocol* (WFMP) which is similar to the *Ipsilon Flow Management Protocol* (IFMP) [NEH+96]. This protocol classifies IP packets in the core network interface and establishes IP flows between terminals and the core network. The defined lightweight WFMP signaling minimizes the wireless overhead and decreases implementation complexity.

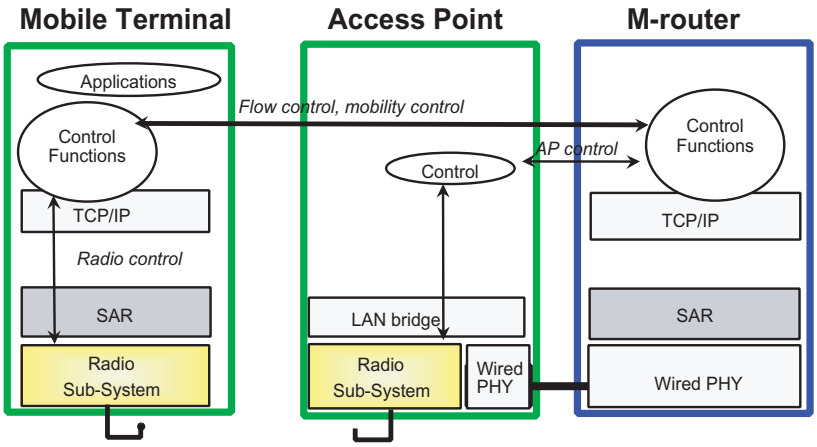


Figure 2.4: System architecture [SALM+98]

2.7 Functional Architecture

The functional architecture of the system is illustrated in Figure 2.5. The main external interfaces are listed in Table 2.1.

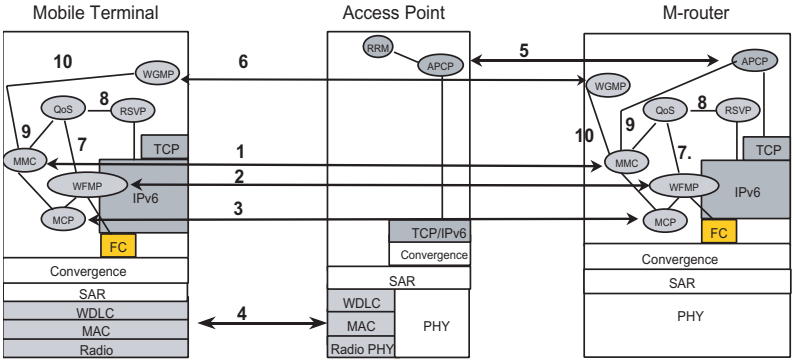


Figure 2.5: Functional architecture and the main interfaces of the system [SALM+98]

The external control interfaces define the logical interface between the MT and the radio access network and between the radio access network and the core network. The external interfaces have to be standardized, if the objective is to define compatible standard systems that can be composed of devices from different manufacturers. In addition to the standard interfaces the system also has several important internal control interfaces which are listed in Table 2.2.

The system architecture includes the following specific functional blocks:

QoS - Wireless QoS Manager: This entity allocates the radio link QoS for the IP packets. The QoS manager has an interface to the *Resource ReSerVation Protocol* (RSVP) module which can give explicit QoS requirements, such as delay and bandwidth for the IP flow. If no explicit QoS parameters are available, the QoS manager assigns the QoS

Nb.	Interface	Explanation
1	MMC - MMC	Mobility management messages between terminal and MR. Mobility management messaging is used as a new terminal registers in the network and in the case of handovers.
2	WFMP - WFMP	Flow management control signaling messages. This is used for establishing and releasing flows.
3	MCP - MCP	Mobile Control Protocol (MCP) provides a reliable peer-to-peer protocol for transmitting WFMP and MMC messages between the MT and the MR. MCP is used for all wireless specific signaling.
4	Radio control messages	Radio control messages are used for transmitting radio link control messages. For instance terminal association and radio flow control signaling is carried here.
5	APCP interface	Access Point Control Protocol (APCP) is used for sending radio link control and radio resource management messages between the APs and the mobile router
6	WGMP - WGMP	Group management control signaling messages. This protocol is used to trace group members in the access network.

Table 2.1: External control interfaces of the system [SALM+98]

on the basis of the differentiated services or on TCP/UDP port information. When the WFMP entity of the mobile router detects a flow, it informs the QoS entity about its evaluation of the flow's packet throughput. This information can be employed to allocate radio link QoS. The QoS manager transmits the allocated radio link QoS values to the WFMP entity which in turn establishes radio flows with the selected QoS.

Mobility Management Controller (MMC): The MMC entity is responsible for terminal mobility management. The MMC entity of the mobile router has a database that contains information about the registered terminals and their current location. During terminal registration, MMC can be used to authenticate the users. The MMC entity of the mobile terminal initializes the handover by sending a handover request message to the mobile router. The MMC entity of the mobile router in turn checks the radio resources in the new AP and requests WFMP to establish new radio flows in the new AP and to release the old radio flows.

Wireless Flow Management Protocol (WFMP): WFMP entity manages the radio flows. It detects IP traffic and classifies IP flows. Whenever WFMP detects a new flow, it passes the flow information to the QoS manager which assigns the correct radio link priority for the flow. Next, WFMP establishes the radio flow with the allocated priority. WFMP also allocates IP level flow identifiers. The MR includes a master WFMP which classifies the flows and maintains the data base of all the existing flows while the MT includes only a simple WFMP entity that multiplexes the IP level flow identifiers into the correct radio flows.

Mobile Control Protocol (MCP): MCP protocol transmits WFMP and MMC messages

Nb.	Interface	Explanation
7	QoS - WFMP	This is an internal interface which is used for transmitting flow establishment requests and QoS information between the QoS manager and the WFMP. As the WFMP entity of the mobile router detects a new flow, it sends a query to the QoS manager to find the radio QoS parameters. In a real implementation QoS and WFMP entities can be integrated into a single entity which removes this interface.
8	QoS - RSVP	Wireless specific QoS controller interacts with RSVP module for obtaining resource reservations and converting them into radio resource reservations and radio QoS. The RSVP module requests resources from the QoS manager via this interface.
9	QoS - MMC	This interface is used in the case of handovers. The MMC module informs the QoS entity about the handover and requests the re-establishment of the wireless flows. This establishment request is then forwarded to the WFMP entity.
10	WGMP - MMC	This interface is used to inform WGMP of the membership changes due to handover.

Table 2.2: Internal control interfaces of the system [SALM+98]

between the MTs and the MR. MCP provides a reliable mechanism for transmitting control information. MCP implements a simple retransmission protocol. A separate low layer protocol was added to guarantee reliable transmission of control messages instead of using TCP/IP. This is because TCP/IP does not allow the separation of control messages from other TCP traffic. Therefore, in case of handover the control messages would be mixed with the user data traffic, which causes a significant delay for the handover procedure and re-establishment of connections. The use of MCP allows control traffic to be prioritized over the user data packets.

Flow Compression (FC) block: The MR and MTs include FC entities which are in charge of header compression. Header compression relies on the fact that many header fields remain the same over the life-time of an IP flow. Fields that do not change between different packets of a flow do not need to be transmitted. Flow compression is used only for the classified IP flows. The other IP traffic is sent without compression.

Radio Resource Manager (RRM): Each AP has an RRM entity that manages its radio resources. In the system, WFMP must send a resource query to RRM each time a new flow is established. The query includes the requested radio flow priority allocated by QoS entity and the flow rate estimated by WFMP. Based on this information, RRM decides whether the connection is accepted.

Access Point Control Protocol (APCP): APCP protocol provides a mechanism for transmitting control messages between the APs and the MR. APCP can be located above the TCP/IP stack which guarantees the reliable transmission of control messages. WFMP deploys APCP for RRM queries and for sending flow control information to the radio

sub-system.

Wireless Group Management Protocol (WGMP): This entity is responsible to trace group members in the access network. In order to save bandwidth, we only forward traffic destined to a group to the APs with active members of the group. Therefore we need to trace the location of group members in the network.

2.8 Conclusion

In this chapter, we presented the WAND environment and its main objectives. The main achievement of the WAND project was a demonstration system which served as a proof of concept for the developed technology. The user trials were successfully conducted in office and hospital environments. The project was also quite active in the standardization process of ETSI BRAN group. In fact, ETSI HIPERLAN II standard has inherited most of the WAND system design. Although WAND final demonstrator is based on ATM, it does not mean that it is reserved for ATM. As we saw in this chapter, there is no technical problem to use IP protocol directly on the WAND radio sub-system.

We presented a general overview of an IP based access system as well as its functionality. The main open issues are IP QoS support, IP mobility support, as well as multicast communication support at the access system. These issues are further investigated in the following chapters. Finally, our main contributions were the participation to the overall architecture design of the system, the WGMP entity as well as the description of its interfaces with other blocks [ALSNS98] [SALM+98].

Chapter 3

QoS and Flow Management

QoS support is one of the major design objectives of the proposed wireless IP architecture. The main problem of QoS support in the system is the lack of a homogeneous IP QoS standard. Thus, the first step will be to identify the main IP QoS mechanisms in order to provide appropriate schemes for their support at the access network.

Another problem is due to the connectionless nature of the IP network. Each IP packet contains enough addressing information in order to be routed independently of all others. This transmission scheme does not enable the system to separate different traffics in the radio access network and to derive appropriate radio QoS parameters for them. The only possible way is to monitor the IP traffic and to detect and classify IP packet streams having the same QoS requirements called *IP flows*. The access network can then assign QoS parameters for the detected flow which is essential for multimedia service implementation. Although flow detection and selecting QoS are two independent processes, they are closely related since the motivation behind detecting flows is to determine which flows require better services.

This chapter presents the QoS and flow management principals used in our proposed wireless IP network. The main questions that we tackle in this chapter are:

- How can we detect a flow at the IP level?
- How can we map the QoS requirements of an IP flow into the radio access QoS parameters?
- How can we provide QoS at the radio interface?

We start by the description of the Internet QoS parameters as well as the two main QoS approaches used in the Internet. Then we explain the principals of flow detection at the IP level. The problem of mapping IP QoS parameters into the radio QoS is investigated afterwards. The details of how each QoS requirement is actually implemented at the radio sub-system is discussed next. Finally, we provide a detailed description of the QoS and flow management entities as well as their interactions.

3.1 Internet QoS

Internet QoS can be expressed as a combination of network imposed *delay*, *jitter*, *bandwidth* and *reliability* [FH98]. Delay is the elapsed time for a packet to reach its destination. Jitter is the perceived variation in delay. Bandwidth is the maximal data transfer rate that can be sustained between source and destination. Finally, reliable transmission delivers all the packets in the correct order without any bit errors. Reliability is a property of the transmission system and is affected by the average error rate of the medium and by the congestion state of the network. In the fixed Internet packet loss is caused mainly by congestion. However in wireless networks, both congestion and the unreliability of the media must be considered.

When we assess QoS provisioning in an end-to-end way, traversed transmission links must be examined in terms of the four basic QoS metrics described above. ATM is a solution relying on a homogeneous network with a unified data unit, the ATM cell, and QoS guarantees are achieved with statistical multiplexing. While ATM is the state-of-the-art QoS platform, it seems increasingly unlikely that complete end-to-end networks will be based on ATM. Instead IP will be the real connecting networking layer. IP operates over any data link such as ATM, Ethernet and Frame Relay, and over any physical link such as copper, optical fiber and wireless. Given the low probability of homogeneous end-to-end data link networks, QoS mechanism should be based on IP and separated from the specific QoS mechanisms within one transmission link. This situation is depicted in Figure 3.1.

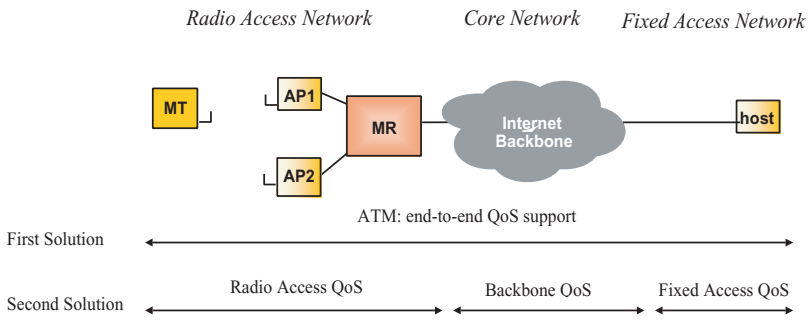


Figure 3.1: QoS solutions [SALM+98]

The Internet has traditionally operated on a *best effort* basis. Best effort service is adequate for legacy data traffic and applications that can adapt to bandwidth and delay variations. However, the type of applications carried over the Internet is expanding. As a result, new end-to-end QoS requirements for the Internet backbone are being set.

IP QoS work is being conducted within *Internet Engineering Task Force* (IETF). This work focuses on two main streams for QoS control, namely *Integrated Services architecture* (IntServ) [Wro97b] and *Differentiated Services architecture* (DiffServ) [BBC+98]. Integrated services architecture tries to provide the QoS in an end-to-end manner like ATM . It allows users to communicate their QoS requirements to routers on the data path by means of a signaling protocol. Differentiated Services architecture specifies the IP header bit usage to differentiate between different QoS classes. The main objective of the DiffServ is to spec-

ify a QoS mechanism based solely on the contents of the IP header fields rather than on an end-to-end signaling protocol. The operation of IntServ and DiffServ are described in more detail in the following sections.

3.2 Integrated Services

The main part of the integrated services architecture is RSVP [BZB+97] [ZDE+93], an end-to-end signaling protocol allowing users to reserve resources along their paths to the sender. In order to reserve resources across the network, RSVP uses the *One Pass With Advertising* (OPWA) mechanism. Senders advertise application traffic characteristics in PATH messages. Routers between senders and receivers modify PATH messages to describe the service they provide (e.g. how much delay they contribute to the overall end-to-end delay). Receivers determine their QoS requirements on the basis of the PATH message contents. The receiver QoS requirements are transmitted to the sender in RESV messages. The information contained in RESV messages is then used by routers and senders to reserve the requested QoS.

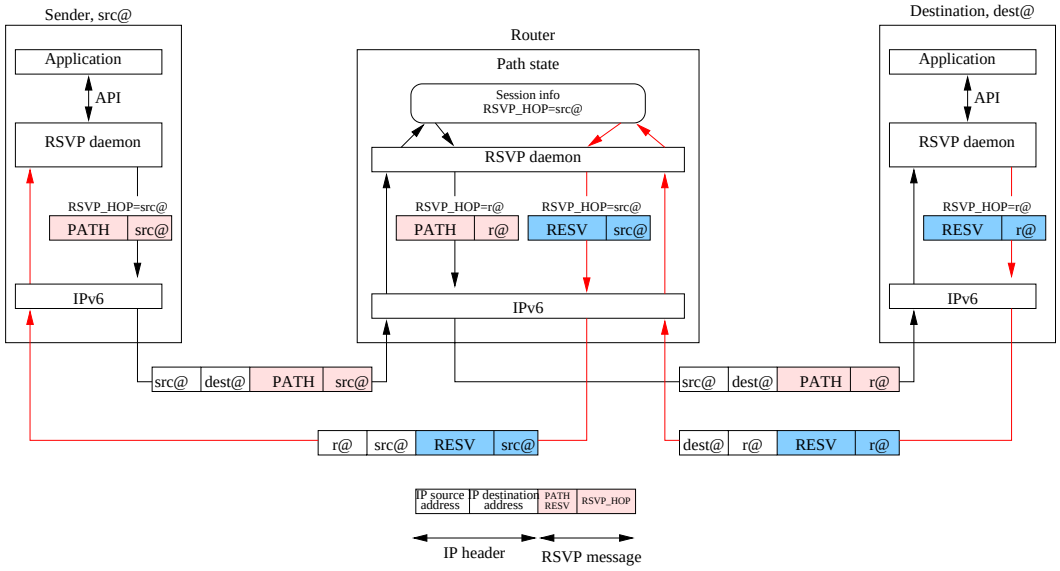


Figure 3.2: RSVP operation

In RSVP, reservations are receiver-driven. As a result, RSVP defines a *session* by its: destination address, IP protocol ID and optionally the destination port. Furthermore, receivers can select a set of senders that can use the resources reserved by the receivers along their path by using a *filter*. In fact, the session specification combined with a *filter specification* (filter spec) defines the set of packets that will receive the reserved QoS. Each filter spec contains the source address, and optionally the source port. The filter specifications associated with a session depends on the reservation style. There are three reservation styles in RSVP:

- Explicit sender selection or fixed filter style,
- Wildcard reservation or wildcard filter style, and

- Shared sender selection or shared explicit filter style

In explicit sender selection reservation, each filter spec must match exactly one sender. Hence the resources are reserved for packets from a single sender to a set of receivers. In contrast the wildcard reservation style reserves resources for traffic from any sender to the session's set of receivers. This means that no filter spec is needed since the reservation is shared over all possible senders. The third alternative reservation style is shared sender selection where the reservation is shared by a specific set of senders. In this case, the reservation request contains a filter spec for each sender.

A PATH message includes a SENDER_TEMPLATE object containing the sender IP address as well as some additional information to identify the sender and a SENDER_TSPEC object specifying the traffic characteristics of the flow. The SESSION object in both PATH and RESV messages contains the session information (destination address, IP protocol ID and optionally the destination port). The RSVP_HOP object in a PATH message contains the previous hop address which is the IP address of the interface through which the PATH message was sent. A RESV message contains a SESSION object, a FILTER_SPEC object defining the filter spec, a FLOWSPEC object defining the desired QoS for each filter and a STYLE object expressing its reservation style. The RSVP_HOP object in a RESV message contains the IP address of the interface through which the RESV message was sent. The main components of PATH and RESV messages are depicted in Figure 3.3.

PATH message

RSVP header	SESSION	RSVP_HOP	SENDER_TEMPLATE	SENDER_TSPEC

RESV message

RSVP header	SESSION	RSVP_HOP	STYLE	FLOWSPEC	FILTER_SPEC

Figure 3.3: The main components of PATH and RESV messages

Upon receiving a PATH message, the RSVP-capable node creates a *flow state* for the sender defined by the SESSION, SENDER_TEMPLATE, SENDER_TSPEC, RSVP_HOP and any other object in the PATH message. A PATH message travels in an end-to-end way therefore, the IP source address of a PATH message must be the sender address while its destination address must be the destination address specified in the SESSION object. In fact, PATH messages traverse exactly the same route as data packets. RESV messages, however, travel hop-by-hop from receivers to senders, along the reverse paths of data flows for the session. In order to be routed correctly to the next hop toward the sender, the RESV messages use the RSVP_HOP object of the flow state maintained in each RSVP-capable node. The IP destination address of a RESV message is the address of the previous hop node, obtained from the flow state. The IP source address is the IP address of the node that sent the message. The operation of RSVP is illustrated in Figure 3.2.

RSVP has a *soft state* approach to manage the reservation states in routers and hosts. Flow state is created and periodically refreshed by PATH and RESV messages. The state is

deleted if no matching refresh messages arrive before the expiration of a timer. States may also be deleted by an explicit TEARDOWN message.

The integrated services architecture proposes three different QoS classes as follows:

- *Guaranteed Service* (GS) [Dee89] guarantees a maximum end-to-end delay, and is intended for audio and video applications with strict delay requirements.
- *Controlled Load Service* (CL) [Wro97a] guarantees to provide a level of service equivalent to best effort service in a lightly loaded network, regardless of network load. This service class is designed for adaptive real-time applications (e.g. applications that can modify their play-out buffer as the end-to-end delay varies).
- *Best Effort Service* (BE) provides no service guarantees.

3.3 Differentiated Services

The differentiated services architecture focuses on a way to provide different levels of QoS without the need for signaling. It is believed that *Internet Service Providers* (ISPs) will not want to use signaling mechanisms such as RSVP to provide fine grained resource allocation. Instead users' needs can be satisfied by offering a range of different services from which they can select. Strict real time per flow guarantees are not envisioned as an objective of the DiffServ, rather the effort is focused on the provision of different levels of best effort service differentiated on timeliness of delivery and drop precedence. A low loss, low delay service will be priced higher than a service that provides worse performance in terms of loss rate and delay.

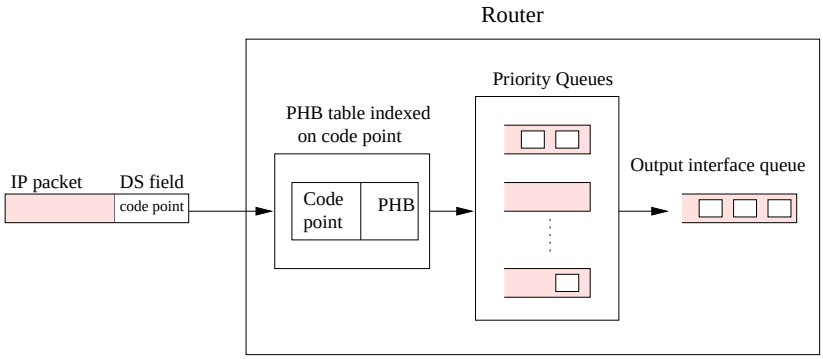


Figure 3.4: Differentiated Services operation

IETF proposes the use of the *Type of Service* and *QoS Class* fields of IPv4 and IPv6 headers respectively to differentiate between QoS services. This field is also called *Differentiated Services* (DS) field. All packets will carry a code point in their DS fields which will be used to select the proper packet handling, or *Per-Hop Behavior* (PHB) that the router will have to provide. PHBs can be seen as a set of building blocks that are used to create a wide range of different services. A configurable table with a one-to-one code point to PHB mapping

will be used in every router in order to allow flexibility and to provide operator needs. A packet has to be classified into proper queues based on its code point. A general view of the DiffServ operation is depicted in Figure 3.4. PHBs are expected to be implemented by employing a range of queue service and management disciplines on a network node's output interface queue. An example is to use a *weighted round-robin* queue servicing or to use a *drop preference* queue management.

Proposals include varying the number of predefined bits for delay and dropping priorities to allow each packet to be handled independently based on the header bits. The standardization of priority bits and PHBs is still underway within the IETF, and there may be different ways to deploy priority bits in the future. Two different PHBs have been standardized by IETF up to now, but their implementations have been left open intentionally in order to allow further experimentations. In general, we can identify three different schemes:

- *Expedited Forwarding* (EF) provides a low delay, a low loss and an assured bandwidth. Such a service appears to the endpoints like a point-to-point *connection* or a *virtual leased line*.
- *Assured Forwarding* (AF) provides delivery of IP packets in four independently forwarded AF classes. Each of these classes has a share of the bandwidth. Within each AF class, an IP packet can be assigned different levels of drop precedence.
- *Best Effort* (BE) which provides no guarantee.

3.4 IntServ versus DiffServ

Table 3.1 provides a comparison between integrated services and differentiated services architectures.

	IntServ	**DiffServ**
Granularity of service differentiation	individual flow	aggregate of flows
State in routers	per-flow	per-aggregate
Traffic classification basis	several header fields	DS field of the IP header
Signaling protocol	required (RSVP)	not required
coordination for service differentiation	end-to-end	per-hop
Scalability	limited by number of flows	limited by number of classes of service

Table 3.1: Comparison of IntServ and DiffServ [DR99]

Integrated services architecture works on a per *flow* basis. An IP flow is a stream of packets that have the same QoS requirements. It refers to the flow of IP packets that belong to the same connection, i.e. the IP packets that are sent between particular applications (port) and between particular hosts (IP addresses). In fact, IntServ tries to establish a connection over the connectionless IP. The use of IntServ over the backbone network may cause a large number of flows which in turn limits the scalability of the scheme. On the other hand,

RSVP state maintenance in routers consumes a lot of memory and its periodic state refresh messages increase the Internet traffic load. Due to these problems, it currently looks like that RSVP is going to be employed for access networks but not necessarily for end-to-end connections through the backbone network. Therefore, it is still important to consider RSVP integration for our wireless access network.

The Differentiated services architecture, on the other hand, tries to overcome the limitations of IntServ by aggregating the individual flows into a few classes. QoS decisions are taken in a hop-by-hop basis and no end-to-end control is provided. This architecture is believed to provide more scalability in the backbone network.

3.5 QoS Support Issues

Up to now, we said that the radio access QoS mechanism must be independent from the backbone QoS mechanism but at the same time it must provide a way to map the fixed network QoS parameters into the air interface. IP implements a connectionless packet data system. The data is carried inside packets, the header of which indicates the correct destination address. This transmission scheme does not enable the system to separate different traffics on the radio access network and to derive appropriate radio QoS parameters for them. The only possible way is to monitor the IP traffic and to detect and classify IP flows. In this way, the network can assign certain QoS characteristics for a flow, which is essential for multimedia service implementation in IP networks. For instance, a particular flow can be prioritized in the router.

Once a flow is detected on the network layer, we need to map its QoS requirements on the radio access QoS. We introduce the concept of *radio flow*. Radio flow is an equivalent concept for wireless transmission links. It is a wireless connection having certain QoS requirements. The wireless IP system has to be capable of distinguishing various IP flows both in the network side and in the air interface. The initial WAND radio sub-system allows different QoS parameters to be allocated to various connection types. Therefore it is essential to specify a mechanism for mapping IP level QoS parameters assigned to an IP flow into the radio flow QoS. Figure 3.5 illustrates the principles of QoS management scheme.

As a conclusion, in order to support QoS at the access network, mechanisms must be proposed:

- to detect and identify a flow at the IP level,
- to map the QoS requirements of a flow on the radio access QoS, and
- to provide QoS at the air interface.

In the following sections, we try to investigate these issues further.

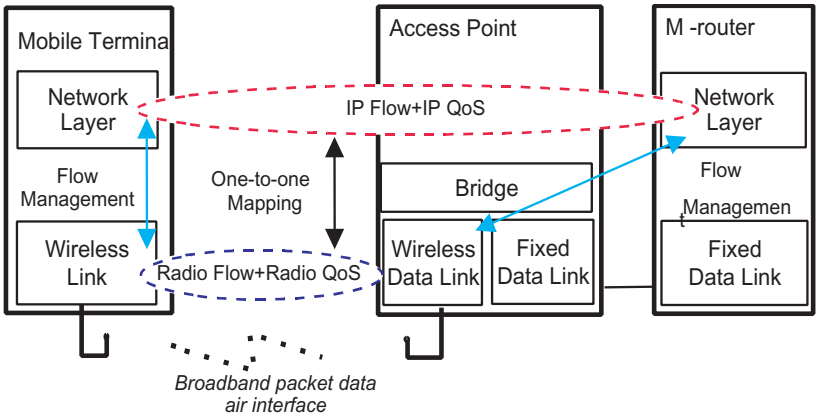


Figure 3.5: Principal QoS management scheme

3.5.1 Flow Detection

Different approaches are possible for flow detection. IPv6 *flow labeling* mechanism may be the primary choice. Flow Label is a 20-bit field in the IPv6 header which may be used by a source to label sequences of packets for which it requests special handling from the IPv6 routers. According to [DH98], all packets belonging to the same flow must be sent with the same source address, destination address, and flow label. If any of those packets includes a *Hop-by-Hop Options* header, then they all must be originated with the same Hop-by-Hop Options header contents. Furthermore, if any of those packets includes a *Routing Header*, then they all must be originated with the same contents in all extension headers up to and including the Routing header. IPv6 flow identifiers are listed in Table 3.2. Note that the Hop-by-Hop option as well as the *Destination Option* headers are normally placed before the Routing header.

Flow Type	**IPv6 Header Fields**
I	Source Addr; Dest Addr; Flow Label
II	Source Addr; Dest Addr; Flow Label; Hop-by-Hop Options
III	Source Addr; Dest Addr; Flow Label; Hop-by-Hop Options; Destination Options; Routing Header

Table 3.2: IPv6 flow identifiers

Therefore we can assume that only packets with non zero flow labels should be selected as flows requiring a special QoS. The problem with this approach is that it is very possible that many real-time applications do not employ the flow label, yet they want more than best-effort service. Another issue in this case is to decide the QoS requirements of the flow, since no QoS information is provided.

Ipsilon (now Nokia Networks) *IP Switching* [NLM96] [NEH+96] [NML97] and Toshiba *Cell Switch Router* [KNE97a] [KNE97b] approaches support several flow detection approaches based on RSVP, traffic volume, and identification of specific higher layer protocols such as WWW and FTP. These approaches use data-driven flow techniques where flows are detected when data arrives for that flow except for the case when RSVP triggers a flow.

In contrast Tag Switching [RDR+97] [RDK+97] employs a topology-driven flow detection. In this case flows are identified based on routing information before any data arrives. This means a flow is able to be associated with a group of routes, a multicast tree, a source-destination pair, an application operating between a given source and destination, or any other policy [RDR+97] [RDK+97]. *MultiProtocol Label Switching* (MPLS), currently being specified by the IETF, is required to support both data-driven and topology-driven flow detection techniques.

As a result, many flow detection techniques can be employed at multiple granularity levels. Hence the choice of flow detection technique will become an implementation issue. In the wireless IP system, the motivation behind detecting flows is to assign them a suitable QoS in the radio level. We do not use a topology-driven flow detection because it does not allow us to sufficiently separate traffic on the basis of QoS requirements due to its coarse level of granularity. For example if all traffic from a given company is grouped together, then if one user is running a real-time video application and another is WWW browsing, their traffic would be treated in the same manner.

Our flow detector must be able to use the QoS information of a traffic to detect flows that need to be carried with non best effort QoS. In this sense, it should operate with both IntServ and DiffServ. We also want to have a flow detection mechanism for long-lived traffics. The motivation behind this is twofold:

- to reduce the delay for high volume traffic, especially over the air interface, and
- to reduce the IP level processing burden on the router.

Due to the desire to provide better than best effort QoS to long-lived flows, it may also be beneficial to add a packet volume based detection scheme to the flow detector. One example of this approach would be for the flow detector to measure the volume of packets with the same flow identifier. When this exceeds a predefined threshold, a special flow could be created for that traffic. The threshold could be the number of packets seen within a given period of time for example if more than 5 datagrams with the same flow identifier are seen in 30 seconds. This mechanism may also help to obtain an estimate of the required radio resources. Essentially, the flow detector should be flexible enough to detect flows on the basis of a variety of criteria.

In case of IntServ, the arrival of RSVP PATH and RESV messages could be used by the flow detector to indicate that a group of packets should receive a new level of QoS. RSVP RESV messages contain a filter spec that identifies the set of packets that should receive the desired QoS. The session information (destination address, protocol ID and optionally destination port) is also present in RESV messages. Therefore, flows can be detected on the arrival of RSVP RESV messages. In this case, all IP datagrams with header fields that match both the session and filter spec details should receive the new level of QoS. In practice the RESV messages can be decoded in the MR to obtain explicit QoS parameters for the RSVP flow. This approach can be used in conjunction with two levels of granularity: one on a per host basis (no destination port provided), and the other on a per application basis (destination port is provided).

An alternative approach to requesting QoS in the Internet is via Differentiated Services. DiffServ sets bits in the IP header to indicate the delay and dropping preferences of that datagram. If this field is zero the traffic should be carried in a best effort fashion. IntServ tends to be used in a dynamic manner, that is, when an application requiring QoS begins, it informs the routers across the network through RSVP messages that resources should be reserved for that specific RSVP session. In contrast, DiffServ tends to be used to preprovision QoS for each of the traffic classes, the network operator wishes to support. This means that when the detector sees a datagram with a given traffic class, it does not need to trigger the detection of a flow but simply to note via which existing radio priority class, the packet should be transmitted. Since the detection decision is on the basis of the traffic class only, this approach can be employed on the granularity of all packets requiring the same level of QoS regardless of source and destination. Indeed, dedicated flows are not required, instead the DiffServ QoS classes can be mapped to the appropriate radio flow QoS parameters and the packets must be sent using this QoS. However, if the detector regularly observes packets between two hosts with their DS fields set, it can trigger the creation of a new flow.

In order to detect long-lived flows, we have to take into account additional criteria for detecting flows, such as transport protocol port numbers. Significant processing time may be required to access some header fields (e.g. the TCP or UDP port number) since all preceding headers need to be parsed first.

Another issue closely related to flow detection that must be considered is how to decide when to tear down or remove flow specific information. This could be based on a timer. In this case, flow state information is removed when no packets matching the flow are seen over a specified time period. The difficulty in this case is selecting the value of the timer. This issue is considered in [SK94]. RSVP flows can be torn down if RSVP update messages are not received, or if an explicit RSVP TEARDOWN message is received. Another approach would be to wait for the arrival of a specific message to indicate the flow is finished. For example the arrival of TCP FIN messages. The problem with this last approach is that it requires processing of higher layer protocols. Hence we prefer a timer based solution for all flows. However for RSVP flows the arrival of TEARDOWN message can also be used to decide whether to terminate the flow.

In summary, in our wireless IP the flow detector must detect flows based on:

- the arrival of RSVP RESV messages, or
- the regular arrival of IP datagrams between two hosts with DS bits set : priority bits + source address + destination address, or
- the arrival of long-lived traffics: source address + destination address + port numbers + protocol id.

Once a flow is detected, the flow detector must create a flow identifier and maintain the flow information as well as the flow identifier in a database that we call *Active Flow Database*. Each flow has a *Status* filed in the data base indicating how it is created. If after a certain timeout, no packets arrive for a flow, it is deleted automatically. In case of RSVP, the arrival of TEARDOWN message will cause the flow to be deleted from the active flow database.

Next section discusses the radio link QoS classes and the mapping between IP QoS parameters to radio QoS classes.

3.5.2 Radio Link QoS Classes

In wireless transmission links, multiplexing different services into the medium requires consideration of four QoS parameters: bandwidth, delay, jitter, and reliability. Bandwidth is the first requirement for QoS driven services in order to support the requested traffic parameters. In the wireless link, the main objectives are efficient channel utilization while maintaining service specific QoS for IP traffic. This is translated into a need to have a rate control scheduling entity in the AP. The AP Scheduler should know the requested average and/or peak bandwidth of those connections for which the radio flow is to be established. In this way, the Scheduler can guarantee the satisfaction of bandwidth on demand.

In order to fulfill the other QoS requirements, we define three priority classes in the radio sub-system. The first class has the highest priority. It corresponds to applications with high delay constraints. These kinds of applications are normally more resistant to packet losses such as voice. The second class has less priority than the first class. It is suitable for applications that have a medium delay and reliability requirements such as video. The least priority class is for the data which can support a high delay but it demands a low loss rate.

Table 3.3 presents the mapping from IP level QoS into the radio access network specific QoS. In case of port-based flow detection, we provide the QoS as a function of that specific application. For example, FTP needs a lot of bandwidth but doesn't have critical real time requirements. On the other hand, TELNET doesn't need much bandwidth but is adversely affected by high delay.

Priority Class	**IntServ**	**DiffServ**	**Well-known Ports**
1st class low delay/high dropping	Guaranteed	Expedited Forwarding	vat
2nd class medium delay/medium dropping	Controlled Load	Assured Forwarding	telnet, http
3rd class high delay/low dropping	Best Effort	Best Effort	ftp

Table 3.3: Network QoS mapping into radio access QoS [SALM+98]

The MR will function as the central intelligence point of the radio access network detecting flows, classifying them and mapping network QoS requirements into radio QoS classes. The details of how each QoS parameter is implemented at the radio access network is the subject of the following sections.

3.5.3 QoS Strategies in Radio Access Network

3.5.3.1 Header Compression Scheme

In a wireless environment, the bandwidth is a scarce network resource. Hence, it is important to minimize the volume of traffic being transmitted over the air interface. This can be achieved via IP and transport header compression. Header compression relies on the fact that many header fields remain the same over the life-time of a traffic stream. Fields that do not change between packets do not need to be transmitted.

Compression is even more important when carrying IPv6 datagrams compared to IPv4 datagrams due to the significant increase in header size. The IPv6 base header alone is 40 bytes. The radio overhead is minimized by compressing the IP headers of detected flows. The compression is performed between the MR and the MT. For this purpose, they include specific *Flow Compression* (FC) entities. The IP header compression is efficient for flows as the IP source and destination can also be identified from the flow identifier. The receiving end can look up the flow identifier and decompress the IP header accordingly.

However, it is also important to keep in mind that the bandwidth savings of the proposed scheme are a trade-off against the processing requirements on the MT and the MR, hence the complexity of the compression algorithms must also be considered. We recommend the use of the IPv6 header compression approach [DNP99] developed in the IPng IETF working group since this is more likely to gain wider use in the IPv6 backbone. If IPv6 header compression is employed in the backbone, this will reduce the compression processing requirements on the MR, because it will receive packets already compressed.

3.5.3.2 Error Control and Queuing

Delay and Jitter are primarily affected by the traffic scheduling and queuing over the wireless link. The initial WAND radio sub-system supports different QoS classes in a per-connection basis. Each connection is queued separately. The Scheduler assures that the QoS requirements of each connection is fulfilled. An IP flow represents a *soft* connection between two hosts at the network layer. By soft connection, we mean that the connection needs to be refreshed periodically in order to persist. A radio flow is the equivalent concept of the IP flow at the radio sub-system. In the radio sub-system, the radio flows are queued separately. They are also grouped into three different priority classes. In order to place the packets in the right queue, the Scheduler needs to know the flow ID and priority class of the incoming packets. Also, in order to take the delay and jitter requirements into account in choosing the packets to be sent, the Scheduler should know the maximum allowed delay of the packets at RAN layer. Therefore it must keep a time stamp for each packet.

The queuing strategy is therefore based on radio flows such that each radio flow has its own queue. Based on the flow ID, the right priority class can be chosen as well as the queue where the packet is placed. This approach is needed because the Scheduler has to be able to differentiate the connections and their QoS requirements. It also provides a transparent way to support integrated services at the radio access network since each radio flow corresponds to a connection at the radio sub-system.

As stated before, DiffServ is used on the granularity of all packets having the same QoS requirements. These packets can have different sources and destinations. Even though they do not belong to a detected flow, they have some QoS requirements and the radio access network must provide some mechanisms to provide QoS to these packets. We use three default radio flows in the radio sub-system in order to facilitate the support of DiffServ. Each of these default radio flows corresponds to one of the radio priority classes. These default radio flows have also their own queues. According to their DS fields, the packets are forwarded to the right queue.

Reliability over the wireless link requires error control which is typically provided via coding and retransmission. *Forward Error Correction* (FEC) uses a coding scheme for both error detection and correction which imposes constant overhead over the applied data. The erroneous packets are not retransmitted, instead the receiver tries to correct the errors by the help of the error correcting code. *Automatic Repeat reQuest* (ARQ) only uses an error detecting code. In case of error, a packet is retransmitted. ARQ is feasible as long as the channel bit error rate is not too high and the retransmission delay is admissible. *Hybrid ARQ/FEC* techniques take the advantages of the two approaches. If the errors in a packet can not be corrected by the error correcting code, a retransmission will be demanded.

The Scheduler needs information about the reliability mechanism usage for each radio QoS classes. The original WAND system uses an ARQ scheme for non-real-time traffic. In our wireless IP system, we propose three different reliability mechanisms. These mechanisms are based on a hybrid approach. Depending on the radio priority class, we use a limited ARQ, an unlimited ARQ or no ARQ at all. In case of real-time applications, a retransmission does not make sense if the deadline of the packet has been expired. That is why the first priority class does not use an ARQ scheme. FEC usage, on the other hand, can be fixed and used for all packets regardless of their priority class. Figure 3.6 provides a summary of our proposed queuing and reliability strategies at the radio sub-system.

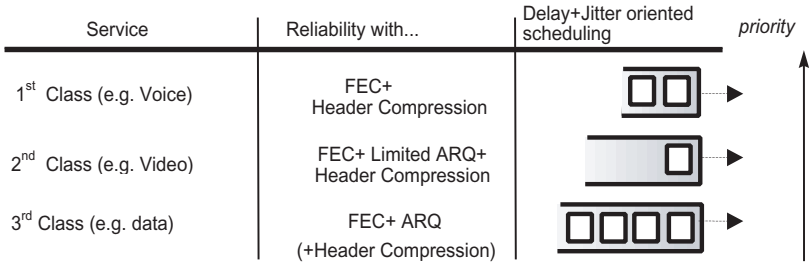


Figure 3.6: Wireless QoS driven queuing and error control strategy [SALM+98]

3.5.3.3 Scheduler

Wireless environments require a scheduling algorithm that is efficient and aware of the QoS and traffic characteristics of the system. The scheduling algorithm has an important role in controlling the flow of the packets over the band-limited wireless channel. Used together with *Connection Admission Control* (CAC) and resource allocation, scheduling can be used to guarantee the satisfaction of different QoS requirements for different traffic types. Admis-

sion control and resource allocation operate at the time when the connection is established, deciding whether new connections can access the channel. The Scheduler decides which packets should be placed in each MAC frame. The scheduling algorithm should aim to provide the following properties:

- QoS requirements satisfaction; it must be able to maintain the delay and loss requirements according to the traffic contract.
- Statistical multiplexing gain; it must be able to smooth or to take into account the effect of each traffic on buffer occupancy in order to avoid congestion.
- Efficient use of bandwidth; it must be able to utilize the resources allocated to an application during its idle time since some applications like WWW browser may not be sending packets all the time.
- Efficient rate control; it must be able to drop the priority of applications producing more traffic than expected and thus breaking their traffic contract.

According to the proposed queuing scheme and the identified characteristics of a scheduling entity in wireless systems, we propose the following mechanism:

The Scheduler prioritizes the packets according to the three defined priority classes. Class 1 has the highest priority and class 3 has the lowest priority. The scheduler begins to allocate packets pending in the class 1 queues. Inside the priority class, prioritizing between packets/flows can be made according to delay requirements i.e. choosing the packet that has the least *lifetime* left and so forth. If there is still space in the MAC frame when all of the priority class 1 packets have been allocated, the Scheduler begins to allocate packets from priority class 2 queues in the same way as in the class 1 case. After all the class 2 packets have been allocated, the Scheduler allocates class 3 packets into the free slots of the MAC frame. The class 3 queue operates in a *First In First Out* (FIFO) fashion while the other two queues operate in an *Earliest Deadline First* (EDF) policy. The scheduling ends when all of the packets have been allocated or when the MAC frame is full. Parallel to this, the flows consuming less bandwidth than allocated must have higher priority. This can be taken into account by using traffic policing functions such as *Token Bucket*.

3.6 QoS Management Scheme

The QoS manager's main task is to map the fixed network's QoS parameters to radio QoS and communicate with the radio resource manager. In practice this means mapping explicit QoS values to radio priority queues. The QoS manager has to know some statistics of the flow in order to proportion it to the available radio bandwidth. With this information, the QoS manager can prioritize different flows. These flows can be treated differently from each other, and QoS can be implemented by multiplexing these flows on the basis of their QoS parameters. These parameters can be explicit values such as peak cell rate or bandwidth requirements or simply information about their preferred class of service. Each case depends on the mechanism employed to determine the QoS parameters [SALM+98] [MLLV98].

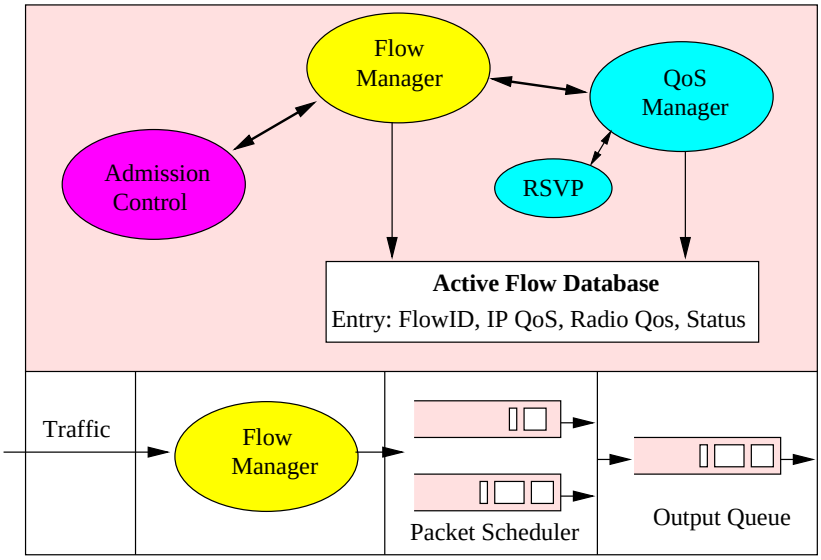


Figure 3.7: Flow detection and QoS management in MR [MLLV98]

MR's QoS manager has more functionalities than the MT's QoS manager, because flow establishment is performed at the MR. Figure 3.7 provides a schematic view of the QoS manager and its interfaces to other entities in an MR. Flow manager (WFMP) is the entity that monitors, detects and classifies packets into IP level flows. Once a flow is detected, the flow manager must contact the QoS manager in order to check the associated IP and radio QoS parameters. The QoS manager has an interface to RSVP in order to find the right priority class for an RSVP flow. It has also the necessary functionalities to understand the meaning of the DS bits and to map them into the right QoS class. It also has the QoS information of different TCP/UDP ports which are reserved for certain applications. Using this information, it can map the detected flows into their corresponding IP and radio QoS.

There are two connection admission control functions:

- *Fixed network Connection Admission Control* (FCAC) which refers to resources in the MR - AP link
- *Wireless network Connection Admission Control* (WCAC) which refers to radio resources in the MT - AP link.

FCAC can be handled by standard IP level CAC mechanisms if necessary like when an RSVP entity requests a guaranteed QoS. The WCAC is performed in a centralized way within the MR on the basis of the information provided by the involved AP. In fact, each AP has an RRM entity in order to control its radio resources. The flow manager transmits the requested radio priority, allocated by the QoS manager, and its estimated flow rate to the WCAC entity which in turn contacts the RRM entity of the corresponding AP. Based on the information provided by the flow manager, the RRM decides whether the connection is accepted or not. In Figure 2.4 in Chapter 2 the flow manager is represented by the WFMP entity and the WCAC block has not illustrated.

After successful admission control, the IP flow information is updated in the active flow database of the MR. Each entry of the active flow database contains a flow ID, its QoS at the IP level, its corresponding radio QoS and the status of the flow representing how the flow was detected. The details of how the QoS manager obtains the necessary QoS information in each case are discussed in the next sections.

3.6.1 Methods to Obtain QoS Information for a Flow

3.6.1.1 Integrated Services

We have two different scenarios to obtain the QoS information of a flow in case of IntServ:

- The WFMP has already detected a flow before the RSVP entity receives the reservation request for that particular flow.
- The RSVP entity receives the reservation request before the WFMP detects the flow. In this latter case, RSVP should trigger the WFMP. This can be done via the QoS Manager.

In both scenarios, the procedure is as follows:

1. RSVP messages use protocol number 46, and that is how reservation messages can be separated from other traffic in the MR.
2. These messages will be delivered to the RSVP entity, which handles the messages on the basis of message type.
3. The RSVP entity talks with the QoS manager which calculates the right priority class for the flow.
4. The QoS Manager asks the WFMP to establish a flow with the appropriate QoS.
5. If the WFMP has already detected a flow, it checks to see if the QoS requirements of the flow have been changed. If they have been changed, the WFMP asks for resources from the RRM. It also updates its active flow table in order to indicate that the flow is signaled by RSVP.
6. If the WFMP has not detected the flow before, it creates the flow and makes an entry for the flow in the active flow table. Then the WFMP asks for resources from the RRM.

RSVP PATH and RESV messages are sent periodically and thus can also be considered as refresh messages. These messages should not trigger a new flow, but only refresh the existing flow information. This is done in the WFMP entity of the mobile router in the following way:

1. RESV refresh message triggers the QoS manager to send a request to the WFMP.

2. The WFMP checks from the active flow table if it has already signaled an RSVP flow for that particular data flow.

3. If the flow already exists, the WFMP sends a confirmation message and performs no other actions.

3.6.1.2 Differentiated Services

In the case of DiffServ, the procedure of obtaining the QoS information is as follows:

1. When a data packet arrives with priority bits set, the WFMP informs the QoS Manager about these bits but only if these packets are detected regularly enough between two hosts.

2. The QoS Manager includes functionality that understands the bits, and maps these bits to the appropriate radio QoS class.

3. The new flow parameters are added in the active flow table.

Note that if the WFMP can not detect a flow for packets including priority bits (not enough packets per second), it should not give any special treatment to these packets. Otherwise, it might get overloaded in the case of huge volumes of occasional packets.

3.6.1.3 Well-Known Ports

There are many *well known* TCP/UDP ports. These kinds of ports can have very different characteristics. For example FTP is a typical application needing a lot of bandwidth, but it does not have critical real-time requirements. Therefore, it is most adapted to the third radio priority class. On the other hand, TELNET does not need much bandwidth, but is adversely affected by high delay. Its QoS requirements match with our second radio priority class.

A question that needs to be considered is whether the network administrator should be able to configure ports that receive special handling. It may be desirable to configure the classification on the basis of what the customer company needs. Some companies may use multimedia applications much more aggressively than others. Also, some companies may use their own applications that should get most of the bandwidth (e.g. banks). These kind of special treatments are possible, if the QoS Manager is a separate functional entity that can be updated easily.

3.7 Flow Management Scheme

Figure 3.8 illustrates the flow management procedure. An overview of the proposed flow management approach is as follows [SALM+98]:

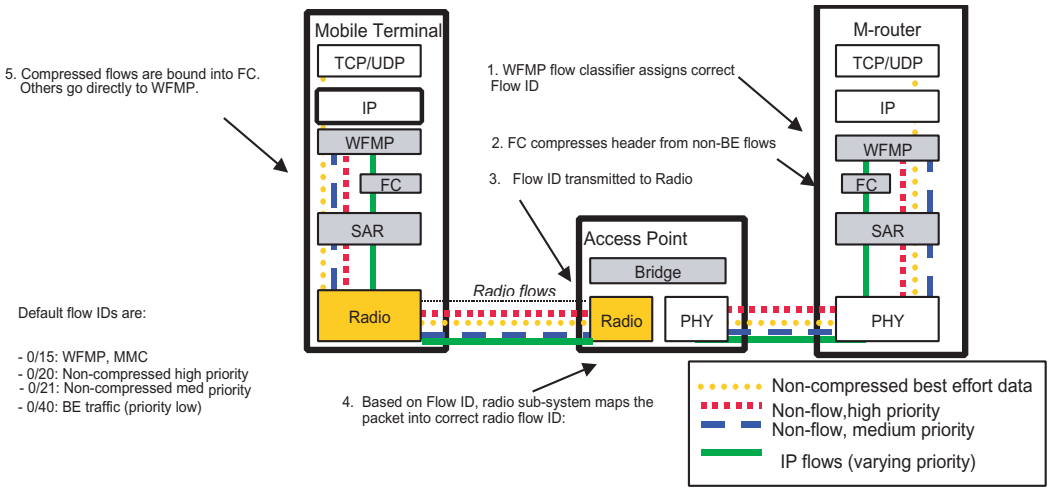


Figure 3.8: IP and radio flow multiplexing scheme [SALM+98]

The WFMP entity in the MR monitors the incoming traffic continuously. If the amount of packets between certain hosts (ports) exceeds a certain threshold value or if an RSVP RESV message is received, the WFMP detects a new flow. The WFMP entity of the mobile router allocates a unique flow identifier for each detected flow. It then starts marking the packets that belong to this flow with its allocated flow ID. The flow ID is a 24-bit identifier utilized to separate packets belonging to IP flows in the access network. Consequently, the use of flow identifiers creates *virtual connections* through which IP flows are packed across the access network. The motivation behind the use of a 24-bit identifier is to have compatibility with WAND ATM-based radio. In fact, the flow identifiers correspond to ATM VPI/VCI identifiers. On the other hand, MPLS also uses a 24-bit label. WFMP is in charge of managing the flow identifiers. Both the MT and the MR have WFMP entities which communicate peer-to-peer over the wireless link. The WFMP actually provides the convergence layer functionality. The MR WFMP detects flows, allocates flow identifiers and informs the MT WFMP of the assigned ID value.

Next, the packets belonging to a flow are passed to the AP via FC block using the allocated flow ID. The FC entity compresses the IP header of the packet and copies the flow ID to the resulting packet. At the receiving end, the peer FC entity can detect the correct source by decoding the flow ID and decompressing the IP header before the packet is passed to the upper layers. It has to be mentioned that only detected IP flows are compressed.

The system offers three default flow identifiers corresponding to three default radio flows per terminal, one for each radio priority queue. These default radio flows were defined to facilitate the support of differentiated services. There is also a predefined flow identifier used for control messages coming from the MMC and the WFMP. This default flow is specially defined to support inter-subnet mobility. As stated before, we only detect a flow for packets with differentiated bits set, if there are enough packets per second. If these packets are not detected regularly enough between two hosts, the WFMP does not trigger a flow. Hence, these packets will be transmitted using the default flow IDs. In the defined scheme, the MR or the MT can look at the priority bits of a single IP packet and send it with the corresponding

default flow ID. As soon as an IP flow is detected, the packets will be switched into a separate radio flow with a specific QoS and with IP header compression. The IP packets that do not belong to any flow are marked with one of three default flow identifiers. These packets must not pass through the FC since in this case IP packets from multiple different sources are multiplexed into the same radio flow, making IP header compression impossible.

In the radio sub-system, each radio flow is identified by a radio flow identifier. In order to minimize the overhead in the radio channel, the 24-bit network flow identifier is compressed into an 8-bit radio flow identifier at the AP. The AP performs IP flow-radio flow multiplexing mapping between network flow ID and radio flow ID. The AP allocates the radio flow ID values in a per MT basis. In the WAND radio, each MT and each AP is identified by a link layer address. Three default radio flow *pipes* exists for the non-compressed traffic also in the air interface. The default network level flow IDs are mapped into the corresponding hard-coded radio flow IDs. Compressed IP flows are switched into dedicated radio flows. Different flows are separated in the air interface using their radio flow identifiers and terminal link layer addresses. In the access network, each radio level flow can be identified by the triple MT link layer address, AP link layer Address and radio flow ID.

In the MT, the radio sub-system converts the received radio flow ID into the corresponding IP level flow ID value and passes the packet to the SAR layer that reassembles the data into IP packets still maintaining the flow ID information. The compressed flows are then passed to the FC entity which identifies their flow IDs and decompresses them into IP packets. The traffic of the default flow IDs is passed directly to the WFMP.

3.8 Conclusion

We proposed a complete architecture to provide QoS in an IP based access network. In this architecture, the radio access QoS mechanism is separated from the backbone QoS technique. At the IP level, QoS requirements are expressed using either integrated services or differentiated services. In the differentiated services architecture, each packet contains the necessary information for its QoS requirements in its DS field and therefore can be handled independently of the other packets. The granularity of service differentiation is based on all the packets having the same QoS requirements and is independent from a specific source or destination. Integrated services uses RSVP as a signaling protocol in order to reserve resources between a particular sender and receiver or between two specific applications. Thus RSVP can be seen as an end-to-end control protocol creating *soft connections* where resources are reserved between particular hosts or applications in an end-to-end way. In this sense, RSVP is similar to ATM with the difference that reservation states have to be periodically refreshed (soft states) in contrast to ATM where the reservations persist during the lifetime of a connection (hard state).

In order to separate different IP traffics at the radio access network and to derive appropriate QoS parameters for them, we introduced the concept of IP and radio flows. These concepts enable us to handle the Internet traffic without requiring extensive data processing in the APs at the TCP/IP level. In this scheme the radio flows are always mapped with a certain QoS value which is derived from the IP QoS parameters.

Three priority classes are defined at the radio link layer. At the radio sub-system boundary, each IP QoS class must be matched into one of the radio sub-system priority classes. This is done by a flexible mechanism mapping the fixed network QoS classes into the radio priority classes. Real-time traffic is handled with the highest priority. Flows which have medium delay constraints are scheduled as medium priority traffic while all the other packets are handled as best-effort traffic.

At the radio link layer, radio priorities are implemented by different queuing and reliability mechanisms. Each radio priority class has its own error control mechanism depending on its delay constraints. The choice of the best error control strategy for each radio QoS class is an interesting research area. The best strategy can not be found without further analysis and simulations. The main questions that arise here are:

- What is the best queuing and scheduling strategies at the radio sub-system? We proposed the use of a priority based queuing system. A scheduler entity is also responsible to place each packet in its corresponding queue. It also decides which packets are going to be transmitted first depending on the delay requirements of the packets. Further investigations are necessary in order to find out the best queuing and scheduling strategies.

- What is the best ARQ mechanism for each radio priority class? We proposed the use of an ARQ scheme in order to overcome the packet losses at the link layer but we did not emphasize a specific ARQ protocol. Further investigation is necessary to find the best ARQ strategy for the radio access network. This issue is studied in [MLLV98].

- What is the best FEC mechanism for each radio priority class and does a fixed coding scheme suit for all radio QoS classes? We proposed the use of FEC for all radio priority classes but we did not specify a coding scheme. Furthermore, the use of FEC is considered to be fixed for all classes. These issues are further investigated in chapters 6 and 7. Chapter 6 focuses on the effect of coding over the delay and the loss rate. Chapter 7 compares the performance of a fixed coding scheme with an adaptive coding scheme.

Finally our contributions in this chapter are the participation at the early specification of the QoS and flow management architecture and its general framework [ALSNS98]. Further details in this subject can be found in [SALM+98] and [MLLV98].

Chapter 4

Mobility Management

Mobility is an inherent part of the system. The specified mobility management scheme must provide full terminal mobility within the access network as well as between different networks. The radio access network mobility scheme deploys the handover mechanism already developed in the WAND system while the core network mobility management utilizes IPv6 mobility functions. Unlike in wireless ATM, the handovers can be hard and lossy due to the connectionless nature of the IP network. An important issue which is the focus of this chapter is the effect of mobility on QoS. If a user moves within the same IP subnet (intra-subnet mobility), the mobility is transparent to the network layer and the preservation of the QoS guarantees depends only on the available resources in the access network. If a user moves to another network (inter-subnet mobility), the QoS preservation depends on the IP level QoS mechanism and the backbone resources as well as the available resources in the access network. In case of integrated services where the reservations are based on the source and destination IP addresses, some problems may arise for inter-subnet mobility.

This chapter presents the mobility management scheme used in our proposed wireless IP network. We present an overview of mobile IP. Then, we explain the basics of intra-subnet and inter-subnet handovers. We encountered several problems between RSVP and mobile IP when optimized routing is used between fixed and mobile hosts. We provide a solution to these problems for unicast as well as multicast communication scenarios.

4.1 Mobile IP Overview

Internet transport layer protocols (TCP/UDP) assume that a host's IP address is fixed. This address is used for identifying the host and for routing purposes. In IP, routing is based on the network prefix in a packet's destination IP address. In order to be able to communicate in spite of its movement, an MT must change its IP address whenever it moves to another network. However in this case, the MT would not be able to maintain its transport and higher layer connections because of its address change [Per98].

Mobile IP provides a mechanism called *tunneling* for IP to maintain two addresses per MT, a fixed IP address for identification called *home address* (ha) and a variable one for routing called *care-of-address* (coa). The home address remains fixed regardless of the MT's

position in the Internet. The care-of-address varies as the MT moves around different networks. A network having the same network prefix as the MT's home address is called *home network*. This is the network where the MT has been registered. A router in the MT's home network which is responsible for datagram delivery to the MT is called its *Home Agent* (HA). This router maintains current location of each MT registered in the home network. Any network other than the MT's home network is a *foreign network*. Any node with which an MT is communicating is refered to as a *Correspondent Node* (CN), which itself may be either mobile or stationary.

When the MT moves to a new IP domain, it will gain a new care-of-address. By default, the HA intercepts all packets on the home network addressed to the MT's home address and tunnels them to the MT at its current care-of-address. This tunneling uses IPv6 encapsulation [CD98] and the path followed by a packet while it is encapsulated is known as a *tunnel*. This scheme causes a triangle routing situation as shown in Figure 4.1.

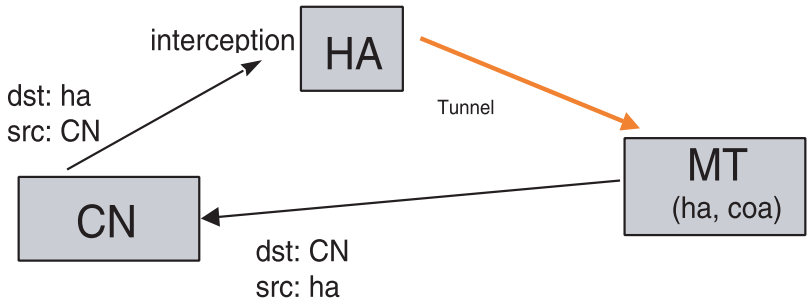


Figure 4.1: Triangular routing in mobile IP

One of the design goals in IPv6 is to minimize the volume of traffic that has to travel through non-optimized routes via the HA. As a result in IPv6, the MT informs not only its HA but also all CNs it is communicating with about its new care-of-address. Once a CN has learned the MT's care-of-address, it may cache it. Each CN can then send traffic directly to the MT in its current location bypassing the HA completely. Similarly, the MT sends traffic to the CN with its care-of-address as the source address. The association between the home address and the care-of-address of an MT is called *binding*. Each IPv6 node must maintain a cache of mobile terminals bindings called *binding cache* [Per98].

The standard mobility procedure in IPv6 is as follows:

- When the MT moves to a foreign network, it must obtain a new care-of-address.
- The MT sends a *binding update* message to its HA. The HA can then act as a *proxy* for the MT intercepting any IPv6 packets addressed to the MT's home address and tunneling each intercepted packet to the MT's care-of-address indicated in its binding cache.
- The MT sends also a *binding update* message to its CNs in order to notify them of its change of address. The CNs in turn add the new care-of-address to their binding cache. Before receiving a *binding update* message, the CNs send data to the MT's old address.

- After binding to the new location, the CNs use direct routing.

This scheme may cause packet losses during handover. When an MT moves to a new network and changes its care-of-address, it can not receive packets destined to its old care-of-address. One enhancement is for the MT to notify its previous router of its new care-of-address. The router will then behave as a HA for the MT and forward any traffic destined to the MT's old care-of-address to its new location. Note, it is still important that the MT informs its HA about its new care-of-address. This will allow traffic from other hosts that are currently not communicating with the MT to reach the MT, via the HA, regardless of its current location. The address information in packets to and from the MT is as follows:

- The CN sends traffic using the MT's care-of-address as the destination address and inserts a routing header containing the MT's home address.
- The MT sends traffic with its care-of-address as the source address and inserts a *home address option* containing its home address.

When a packet arrives at the MT, the routing header is processed which means that the home address becomes the destination address of the packet. Then the IPv6 module processes the packet. Thus all protocols from IPv6 upwards, always see packets arriving to the MT's home address. When a packet arrives at a CN, the home address must be copied from the home address option to the IP source address field. Therefore the use of the care-of-address is completely transparent to higher layers. However, routers on the path between the CN and MT are only aware of the MT's care-of-address. This is because the router header and home address options are only processed at the end-points of the path. Figure 4.2 depicts the use of optimized route between MT and CN.

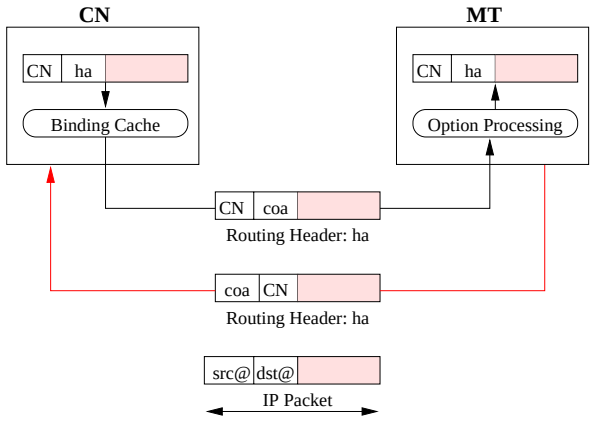


Figure 4.2: Optimized routing in mobile IPv6

4.2 Intra-Subnet Mobility

When the MT moves within the area controlled by one MR, it remains within the same IP subnet and thus retains its assigned IP address. This kind of handover is refered to as intra-subnet handover. The hand-over involves the switching of the control (MMC and WFMP)

and data channels (BE, QoS) from the previous to the new AP which can be realized according to the WAND mobility scheme as discussed in [KHM98]. Connection admission control has to be performed for the new AP. The FCAC is only performed for RSVP flows while the WCAC is performed in a centralized way within the MR on the basis of information provided by the involved AP. If the CAC algorithms grant admission, connections pertaining to the moving MT are diverted to the new link. If the new AP can support lower QoS than requested, the MT has the option of:

- accepting the modified QoS offered by the network,
- converting its QoS traffic to best-effort, or
- completely dropping its traffic.

In such cases, an indication of the change of QoS must be propagated to the CNs. For example, RSVP messages indicating modifications or cancelations of the reservations are propagated to the CNs. Resources allocated to the connections in the old AP are released through the use of explicit signaling instead of leaving the reservations to timeout. The primary goal during the execution of a handover is to preserve the control channels (MMC and WFMP) in the new AP, and if possible the data channels. Since the handover is confined within the same IP domain, no mobile IP specific signaling is exchanged. MMC signaling is exchanged though, to notify fixed network entities of the handover and force the reconfiguration of the access network. The signaling required for the intra-subnet scenario is quite similar to the one in [KHM98] for the present form of the WAND pilot.

4.3 Inter-Subnet Mobility

When an MT moves to an AP connected to a different router than that of its previous AP, an inter-subnet handover must be performed. A new care-of-address is obtained by the MT in the new subnet, in addition to the establishment of its control channels (WFMP and MMC). Address allocation is performed by the *Stateless Address Auto-configuration* mechanism [TN98] which is based on a reserved link-local subnet address and the interface identifier. Using this temporary address, the MT can either obtain a local subnet address using the *Neighbor Discovery* protocol [NNS98] or contact a *Dynamic Host Configuration Protocol* (DHCP) server [BCP00] to obtain an administered address. The newly acquired care-of-address is transmitted to the MT's HA by means of the *binding update* message. The MT also sends *binding update* messages to each CN and also to its previous MR. Thus, any packet received by the previous MR will be tunneled, for the relatively small duration of the handover, to the new network . This is aligned with the objective of keeping handover lossless. Like the intra-subnet handover process, the same CAC procedures are performed. Once again, fixed CAC is only triggered by RSVP.

One of the issues that must be addressed is how to maintain the QoS of the traffic flows to and from the MT when it moves to a new network. The preservation of QoS guarantees during inter-subnet handover depends on the IP level QoS mechanism employed. QoS guarantees in the new network can be maintained by the MT informing the new MR about

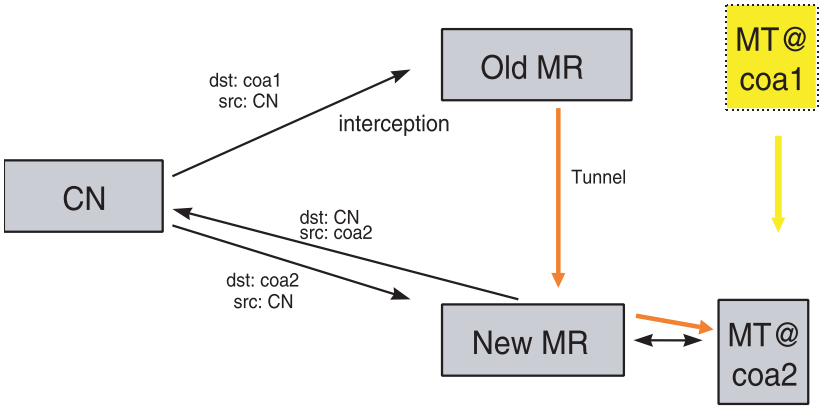


Figure 4.3: Tunneling packets between old and new MRs

the QoS requirements of its active flows. However the maintenance of QoS support within the backbone Internet will depend on whether DiffServ, IntServ or another QoS approach is employed.

When DiffServ is employed at the IP level, each packet contains a series of bits in its IP header describing its resource requirements. Thus, when the MT obtains a new care-of-address, this will not impact the QoS provision across the core network.

The situation becomes complicated in the case of IntServ. IntServ uses RSVP in order to reserve resources across the path between CN and the MT. When the MT moves, the path to its CN also changes, hence the RSVP reservation needs to be updated along the new path. On the other hand, the MT acquires a new care-of-address. This means that the new data packets carrying the MT's new care-of-address will look like they are for a new flow from the intermediate routers' perspective. Rutger's University has developed an approach termed *Mobile RSVP* (MRSVP) [TBA97] to address this problem. Their solution relies on the assumption that the MT provides its *mobility specification* which is the set of locations the MT is expected to visit during the lifetime of its connections. MRSVP then tries to store the required QoS in each location specified in the mobility specification of an MT. The major disadvantage of this approach is the volume of state information in different locations which can become quite high. Furthermore, the assumption that the MT can provide the list of all the networks it is likely to visit is not realistic. The next section presents our solution to the problem of interworking RSVP with mobile IPv6. We discuss the case where the MT has one or more packet flows using the direct optimized route to reach its CN. We consider two types of flows, unicast and multicast.

4.4 The Problem of RSVP Support in IPv6

4.4.1 Unicast Scenario

Mobile IPv6 uses optimized paths between an MT and its CNs. Routing of packets via the HA is typically only a transient phase. When an MT moves to another network, it sends

a *binding update* message to its CN to initiate optimized routing. When the MT moves, it causes a change on the route to its CN. It also changes its care-of-address since it acquires a new care-of-address each time it changes its subnetwork. Both the route and care-of-address changes affect the operation of RSVP. Since establishment of optimized routing is usually a short procedure, we do not look at the phase where packets travel via the HA. However, the operational changes to protocols described here should be initiated as soon as optimized routing is established in order to prevent long reservation setup times.

In RSVP, PATH messages are sent end-to-end while RESV messages are sent hop-by-hop. When the CN is a sender, it will transmit a PATH message where the address details are based on the MT's home address. However, the outer IP header will be modified by the IPv6 binding cache to contain the MT's care-of-address as the destination address. As stated before in Chapter 3, to ensure that the PATH and RESV messages follow the same route, the PATH messages contain a RSVP_HOP object which collects the address of each outgoing interface the message traverses. The CN enters its IP address in the RSVP_HOP object. When the PATH message reaches the first router, it notices that the datagram contains an RSVP message via the protocol ID and pass the PATH message to the RSVP module for processing. The RSVP module creates a flow state and forwards the PATH message on the basis of routing information it retrieves from the routing module. The routing information in case of unicast communication is based upon the session destination address and includes the IP address of the interface to which the PATH message should be forwarded. The RSVP module includes this information in the previous hop field before forwarding it out the correct interface.

Both the CN and the intermediate routers determine the outgoing interface based on the MT's home address. However, the packet will actually be routed based on the MT's care-of-address thanks to the destination address of the outer IP header. Therefore, the routing information stored in the RSVP_HOP object will not be consistent with the route followed. As a result, the RESV messages can not be routed back to the CN due to the incorrect RSVP_HOP information.

For example, we assume a topology as shown in Figure 4.4. We focus on traffic flows from the CN to the MT. At the CN, RSVP PATH messages are addressed to next hop of the HA but the packet is actually forwarded to the next hop router on the path to the MT's care-of-address. At router *R1* the same lookup process occurs. Since the RSVP daemon at the router is unaware of the MT's care-of-address, it will set the previous hop field to be the address of the outgoing interface on the path to the HA. However, it will forward the packet based on the MT's care-of-address and hence towards the next hop router on the path to the care-of-address. This means that flow state will be created at all routers on the path between the CN and the MT's care-of-address, however the previous hop information at each of the routers will be incorrect. Thus when the MT transmits a RESV message, it will not be routed along the direct path from the MT to the CN. Instead it will be routed, based on the incorrect previous hop information, to a router which has no flow state information because it did not receive the PATH message. Hence it is not possible to reserve resources between the CN and MT using mobile IPv6 and RSVP in their current forms.

The routing problem described above occurs when the CN is the sender. When the MT is the sender, the PATH message contains correct routing information. Hence, the RESV message can be forwarded correctly along the reverse path. However, PATH messages contain

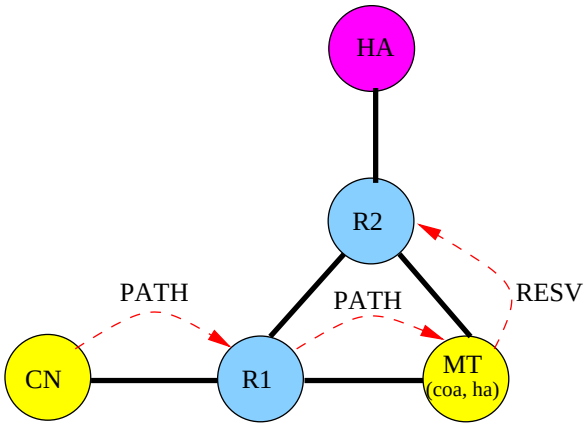


Figure 4.4: Inconsistent hop-by-hop forwarding of RESV messages

the sender's address in their SENDER_TEMPLATE object. When the MT sends a PATH message, the SENDER_TEMPLATE object normally contains the MT's home address but it has to be changed to the MT's care-of-address.

Another important issue is that when the MT moves to a new subnet, its care-of-address changes. In standard RSVP, flow state information is based on the MT's home address. An RSVP session is identified by the triple: destination address, protocol id and optionally destination port. A filter spec along with the session information define the set of packets that receive the QoS defined in a flow spec. It is important to note that the session and filter specs are based on the MT's home address, rather than the care-of-address, since the RSVP module lies above the IPv6 module. However, the data packets to which this information must apply, will be addressed with the MT's care-of-address. Hence, there is an issue of matching RSVP flow state with the data packets belonging to that flow at intermediate routers.

Yet, another difficulty with using RSVP with mobile IPv6 in their current forms, is their applicability to wireless environments where the MT moves to new subnets in real-time. In wireless environments, the speed of rerouting is of major concern. The objectives are to minimize the packet losses, and restore the end-to-end RSVP state as quickly as possible. As a results, the problems that must be tackled are:

- Routing problem, how to route RESV messages correctly back to the CN?
- Flow mismatch problem, how to identify packets belonging to a RSVP session when MT obtains a new care-of-address?
- Flow update problem, how to reserve the required QoS on the new path quickly when the MT moves?

4.4.2 Multicast Scenario

When away from its home network, an MT has two options to receive its multicast data. The first option is through its HA. The HA tunnels the multicast packets to the MT. The

second option is for the MT to join its groups directly from its foreign network. That is, each time the MT moves to a new subnet it must leave its subscribed groups in the old network and must rejoin these groups in the new network. This option allows the MT to receive the multicast packets directly without the HA involvement. As in the unicast scenario, we only consider the situation where we use optimized routes, i.e. when MT receives its multicast traffic directly in its foreign network.

Let's consider the case where the CN sends multicast data and at least one of the receivers is an MT. An RSVP session is identified by the triple destination address, protocol ID and optionally destination port. The destination address of a multicast group remains fixed regardless of the mobility of its receivers. In other words, the MT's exact address and its mobility have no impact on the multicast group address. In fact, an MT making a handover to another network can simply be considered as a new node joining the group. Therefore, the flow state information in each intermediate router is correct and the RESV messages can be correctly routed to the CN.

Now, we consider the case where the MT sends multicast data. IP unicast routing protocols depend only on the IP destination address. Some multicast routing protocols such as *Distance Vector Multicast Routing Protocol* (DVMRP) [WPD88] and *Multicast Open Shortest Path First* (MOSPF) [Moy94] build a multicast tree per source. These routing protocols construct the multicast tree depending on the source and the destination addresses. In DVMRP, multicast routers exchange reverse path distances in order to build a shortest path tree per source. A multicast router forwards a packet if the receiving interface of the packet is on the shortest path to the source. MOSPF takes advantage of the *Link State* database to build a shortest path tree per source. It assumes that each router has the complete topology information. Other algorithms like *Core Based Tree* (CBT) [Bal97] build a single shared tree per group using only the IP destination address for packet forwarding.

Using the home address as the IP source address of a datagram leads the routers executing DVMRP or MOSPF to expect the datagram from the link used to reach the MT's home address. Therefore, sending multicast traffic in a foreign network using the MT's home address as the IP source address means that these routers receive the traffic via unexpected links. DVMRP drops such datagrams because it expects the packet to arrive to the router on the shortest path from the router to the MT's home address. MOSPF forwards the traffic based on an incorrect information. In both cases, some destinations may not be reached. In order to overcome such routing problems, the MT's care-of-address should be used as the source address.

Another issue when the MT is the sender comes from the RSVP which is not aware of mobility. When the MT sends a PATH message, the RSVP fills the SENDER_TEMPLATE object of the PATH message with the MT's home address. It also puts the MT's home address in the RSVP_HOP object of the message. However the IP source address of the outer IP header of the PATH message must be modified to the MT's care-of-address because of the routing problems described above. When the PATH message reaches the first router, the RSVP module of the router creates a flow state and forwards the PATH message on the basis of routing information. If the router uses a source based multicast routing protocol such as DVMRP, the routing information is based upon the session destination address as well as the sender's IP address. Now the problem is that the router will determine the outgoing interface based on the MT's home address. It then puts the IP address of the interface in the

RSVP_HOP object of the PATH message and transmits the PATH message to the next hop. However, the packet will actually be routed based on the MT's care-of-address. Thus, when the PATH message reaches the receivers, the routing information stored in the RSVP_HOP object will not be consistent with the route followed. Hence, RESV messages can not be routed back to the MT. This problem does not occur with centered based tree multicast routing protocols such as CBT.

The problem of flow mismatch also exists for multicast scenario when the MT is the sender. As explained before, the filter spec defines the set of sources to which the QoS reservations must be applied. The standard RSVP uses the MT's home address in the filter spec. Hence, when the MT acquires a care-of-address, the intermediate routers does not recognize the MT. The timing problem also exists for the multicast scenario. The objective is again to restore the reservations as soon as possible in order to have a lossless handover. The summary of problems are:

- Routing problem, how to route RESV messages correctly to the CN?
- Flow mismatch problem, how to identify packets belonging to a RSVP session when MT obtains a new care-of-address?
- Flow update problem, how to reserve the required QoS on the new path quickly when the MT moves?

4.5 Possible Solutions

This section describes solutions and enhancements to the problem of using RSVP with Mobile IPv6. The first solution provides a basic fixed and some optional enhancements to restore reservations in a quick and efficient manner. The second proposal is built partially on top of the first and provides advantages to fast moving terminals inside wireless access networks.

4.5.1 Mobility Enhanced RSVP

Basically, we can resolve the routing problem of RSVP messages between CN and MT either at the end-nodes or at the intermediate routers:

- Solving the routing problem at the end-nodes

  There are two possible ways to tackle the routing problem. One option is to modify IPv6 to provide an interface that allows the RSVP module to look up the care-of-address of an MT. The RSVP daemon at the CN and the MT must be modified to operate on the MT's care-of-address instead of its home address. The RSVP daemon should learn the care-of-address by consulting the binding cache. It should also create the flow state based on the MT's care-of-address. This solution requires the modification of RSVP and IPv6 at the end nodes. An alternative is to modify only IPv6 at the CN and MT. In this case, the mobile IP module needs to become RSVP aware and swap the home address in the PATH and RESV messages with the MT's care-of-address.

- Solving the routing problem at the intermediate nodes

  This approach involves changing the RSVP implementations at intermediate routers. Changes are not required at CNs and MTs. Outer header address information is passed up to the RSVP daemon at each router. Outer header means the IP header transporting the RSVP messages. This allows the RSVP daemon in each intermediate router to learn the mapping between the MT's home address and its current care-of-address. The RSVP daemon should then base its calculation of the RSVP_HOP and the filters on the MT's care-of-address.

We recommend the implementation of an interface in IPv6 which must be used by the RSVP daemon. It only requires to make a small change to CNs and MTs to enable the RSVP daemon to learn the MT's care-of address. In contrast the second solution requires all routers to change their RSVP implementations. Since we can not control the signaling path along the global network with many providers, this approach should be avoided.

As a result, the RSVP daemon can obtain an MT's care-of-address through its interface with Mobile IPv6. It then uses this care-of-address to identify a session. It also uses the care-of-address in the SENDER_TEMPLATE object of a PATH message. The RSVP_HOP object of the PATH message is also calculated correctly based on the MT's care-of-address. Therefore, the RESV messages can be correctly routed back to the sender in both unicast and multicast scenarios.

The above solution only tackles the routing problem but not the flow mismatch problem. When the MT moves and obtains a new care-of-address, all of the intermediate routers will assume this is a new RSVP flow. Hence, there may be situations where the new reservation is denied because the old reservation is still active and consumes resources. This problem could be overcome by introducing a new RSVP object that we call *home address object* to the RSVP messages the MT sends. In this case if the MT is a receiver, it will place its home address in the new RSVP object in the RESV message. This would allow intermediate routers to recognize that the reservation is the same even though the care-of-address has changed. In case of multicast, we also need to identify the packets coming from an MT even if the MT changes its address. Therefore, we also add a new RSVP object to the PATH message which contains the MT's home address.

A keypoint to note is that if some or even all intermediate routers do not recognize this RSVP object, this solution will still work. At those routers that do not understand the RSVP object, the RSVP state with the new care-of-address will be treated as a new independent flow and the previously reserved flow expires later. Note that if the home address was kept as the destination address, and the care-of-address was stored in the new RSVP object, this solution would require all intermediate routers to understand the new RSVP object.

One could argue that the mechanism described above is not required, since IPv6 flow labels in conjunction with the IP source address uniquely identify the traffic flow. However, this will not solve the routing problem of PATH messages. Moreover IPv6 flow labels are optional and hence they can often be zero. If this is the case, the flow mismatch problem still exists. Furthermore, in case of multicast communication, the IPv6 flow label does not allow to identify the traffic. As explained before, the MT must use its care-of-address as its IP source address in order to avoid the routing problems of source-based routing protocols. As

a result, the combination of IPv6 flow label and IP source address does not provide a unique identifier for multicast traffic since the IP source address changes whenever the MT moves to another network.

The last problem is the time required to modify the flow states along the optimal path to the MT's new care-of-address. In standard RSVP operation, PATH and RESV messages are transmitted periodically. However, there can be a significant delay between the MT's care-of-address change and the transmission of the next PATH message. This delay can be avoided by triggering the transmission of a PATH message in CN, whenever a *binding update* message arrives at CN. Again, an interface between mobile IP and RSVP daemon should be used for this triggering.

To summarize the key components of this solution for unicast and multicast communication are:

- At correspondent nodes
  - The RSVP daemon can obtain the mobile terminal's care-of-address from the binding cache.
  - The RSVP daemon places the mobile terminal's care-of-address in the SESSION object of the PATH message.
  - When sending RESV messages, the RSVP daemon use the MT's care-of-address in the FILTER_SPEC object.
- At the intermediate routers recognizing the home address object
  - On receipt of a PATH message, store the mobile terminal's care-of address (standard RSVP operation) and home address (optional for mobility support) in the flow state database.
  - Create a classifier entry based on the mobile terminal's care-of address (standard RSVP operation).
  - When a PATH message arrives with the same home address but a different care-of address update the flow state and filter information with the new care-of address (optional for mobility support).
- At standard intermediate routers
  - On receipt of a PATH message store the mobile terminal's care-of address in the flow state database.
  - Create a classifier entry based on the mobile terminal's care-of address.
  - On receipt of a PATH message with a different care-of address for the mobile terminal, create new flow state information and filters.
  - Remove the old flow state information on receipt of a TEARDOWN message or when it times out.
- At mobile terminals

- Since the MT is reachable under its care-of-address, PATH messages that arrive with its care-of-address in their SESSION objects should not disturb regular RSVP operation.
- When an MT sends RESV messages, the SESSION object must also contain the care-of-address in order to correctly identify the flow on the optimized route.
- The RSVP daemon places the MT's home address in the home address object of a RESV message (optional) for efficient recycling of resources.
- When an MT sends PATH messages, the SENDER_TEMPLATE object must contain its care-of-address.
- The RSVP daemon places the MT's home address in the home address object of a PATH message (optional).

The solution described above means RSVP implementations at CNs and MTs must be changed and RSVP implementations at routers may optionally be changed.

4.5.2 Flow Extension

This section describes an alternative solution to the flow mismatch problem for the fast update of end-to-end RSVP reservations. It mainly refers to the time periods following the occurrence of handover. It is also assumed that, prior to handover, active flows have been established between the CN and the MT. This solution uses the same approach as in section 4.5.1 in order to resolve the routing problem of RSVP messages.

When an MT attaches to a new subnet and acquires a new care-of-address, the MR must intercept and suppress the mobile IPv6 *binding update* message sent to the CN. This prevents the CN from updating its binding cache. This strategy is not applied to the *binding update* message sent to the former MR. This solution can be used to forward the traffic destined to the old location of the MT to its current location. The MT is then capable of receiving datagrams destined to its current IP address as well as the previous IP address. In Section 4.1, we have proposed the use of a packet forwarding mechanism in order to support lossless handover. The flow extension mechanism may be viewed as an extension of the described forwarding mechanism where the packets are forwarded to the new IP domain taking into account their QoS requirements.

It is also possible to suppress the *binding update* messages at the MT without considerable modifications to its mobile IPv6 module. Such approach improves bandwidth efficiency at the radio interface and reduces the complexity of MRs. Disabling the transmission of *binding update* messages at the MT is also adopted in the MRSVP approach.

During handover, the old MR must extend the existing RSVP flows to the new MR. This task is performed by the MMC entity operating within the router. The extension of downlink (CN -> MT) flows is performed by the old MR while the uplink flows (MT -> CN) are handled by the new MR. The new MR needs to receive the characteristics of existing flows from the old MR. For this task, specialized signaling between the MMC entities of the MRs must be introduced. The elongation of flows avoids their invalidation caused by changes in the IP addresses of the endpoints.

The proposed elongation of the CN-MT path causes the route of the communication to be sub-optimal and consequently imposes additional but limited transmission delays. It was shown that consecutive elongation will be needed rarely, as the number of inter-subnet handovers is very small during the lifetime of connections in a customer premises network [VD97]. Most handovers will result in attachments of the MT to the APs in the same subnet. Such mobility events can be handled at the radio link layer without affecting the path beyond the router. The elongation of data paths has already been adopted in wireless ATM technology for handling the connections during handover [ALRR97] [AHK+96] [HPFM98].

In the flow extension approach, modifications are confined to the end-nodes and access network routers. Intermediate routers along the transmission path do not need to be changed. The list of required changes are:

- At correspondent nodes
  - The same basic modifications as described in the Section 4.5.1 are needed. However, since the flows retain their flow states over a long time period, *binding update* messages do not need to trigger PATH messages.
- At standard intermediate routers
  - No changes needed here.
- At wireless access network routers (MRs)
  - The new MR must communicate its IP address to the old MR.
  - The old MR must transmit the existing RSVP flow characteristics to the new MR. To elongate the RSVP state from the old MR to the new MR, an RSVP tunnel [TKWZ00] could be used.
  - The new MR must control and suppress the *binding update* transmission from the MT to its CNs.
- At mobile terminals
  - The same basic modifications as described in Section 4.5.1 are needed. Nevertheless, we do not need to insert home address object in RSVP messages.

Lastly, we are considering how the extension of RSVP flows could be accomplished with existing protocols, such as RSVP, Mobile IPv6 and IP encapsulation. RSVP operation over IP tunnels [TKWZ00] provides a good basis for the implementation of the proposed scheme. The old MR uses regular IP tunnels for forwarding best effort traffic and RSVP tunnels for handling the extension of RSVP flows. The old MR serves as the RSVP tunnel entry point in the downlink direction while the new MR is the tunnel exit point. Roles are inverted in the uplink communication. The tunnel session is a separate RSVP session between the involved routers. Its characteristics are dictated by the characteristics of the flows that need to be extended. The original session (CN -> MT / MR) views the tunnel as a single communication link. The PATH and RESV messages of the end-to-end session are encapsulated at one tunnel end-point and decapsulated at the other. The end-to-end session

and the tunnel session are associated at the entry/exit points of the tunnel. The tunnel may encompass one or more RSVP capable nodes.

The overall scheme is based on the assumption that the new MR is aware of the existence of RSVP flows and thus, suppresses only the *binding update* messages for active RSVP flows. When the entire set of RSVP flows is terminated, the new MR allows the propagation of the *binding update* message to the fixed network. This restores the optimal communication between the MT and the CN regarding best effort traffic.

In a multicast scenario, if the MT changes its subnet, it must rejoin its groups in the new subnet. Now, if the new subnet already has members of the MT's subscribed groups with the same reservations, the MT can receive the data without any delay. If this is not the case, the MT can receive data from its old MR by using an RSVP tunnel. The new MR knows about the MT's subscribed groups and also about the presence of all groups and their reservation styles in its local network. Therefore, instead of trying to graft a path to the multicast tree, the new MR asks the old MR to forward the traffic destined to a group via an RSVP tunnel in the case its local network has no member of the group. The same thing happens, if the new MR has a member of the group but not with the specified reservation style. Now, if the MT is the sender of the multicast traffic (uplink direction), we always pass by the RSVP tunnel to reach the old MR.

4.6 Evaluation of the two approaches

As stated before, the minimal solution to the problem of IPv6 and RSVP integration requires the modification and the interfacing of the RSVP daemon and the IPv6's binding cache at CNs and MTs. This solution requires less changes when compared to an approach that tries to fix the routing problem at intermediate routers. It is important to note that interfacing IPv6 and RSVP requires changes to both standards. For advanced solutions, where performance and smooth handovers in wireless environments are important, we have proposed two solutions:

1. Triggers/Objects: PATH messages are triggered on the arrival of *binding update* messages and home address objects in RESV and PATH messages enable intermediate routers to recognize flows and to reuse resources even when the MT's care-of address changes.

2. Flow Extension: This approach keeps the reservation unchanged until it reaches the wireless access network.

A qualitative comparison of the two approaches is shown in Table 4.1. A quantitative evaluation of these advanced solutions depends on different parameters such as traffic characteristics and network topologies and is subjected to future investigations. Although a minimalist solution would enable the operation of RSVP over mobile IP, we strongly recommend the use of one of the solutions that support fast re-establishment or preservation of the reservations when mobile terminals move. Only such enhancements can guarantee good performance and uninterrupted operation.

	Triggers/Objects	Flow extension
Changes to CN	yes (needed for minimal solution)	yes (needed for minimal solution)
Changes to intermediate routers	yes (RSVP object extension)	no
Changes to MR	no (forwarding of late packets is also an option here)	yes (interception of *binding update* messages, flow forwarding)
Changes to MT	yes (needed for minimal solution)	yes (needed for minimal solution)
Changes to HA	no	no
Supports multicast delivery	yes	yes
Bandwidth efficient	yes	yes
End-to-end delay	always shortest path (but re-establishment of resources requires a round-trip)	only a slight increase in delay
Lossless handover	yes (with forwarding of late packets)	yes
Handover delay	round trip	faster
Implementation complexity	moderate	higher

Table 4.1: Comparison of different solutions for the problem of RSVP support in Mobile IPv6

Without quantitative evaluation, we can just observe that Triggers/Objects is a quick and simple solution that might be able to provide sufficiently good service. The flow extension approach is a little more complex but has the advantage of faster deployment. In multi-provider environments where we cannot control the whole path end-to-end, a solution that modifies only CNs, MRs and MTs has a big advantage. We should also mention that a combination of the two enhancements is possible and useful for large wireless networks and roaming services.

4.7 Conclusion

We presented an overview of the mobility management mechanism used in the wireless IP system. Other issues like address configuration and terminal authentication which are closely related to mobility were not presented here. [HPFM98] and [SALM+98] provide more information about these issues. We also studied the effect of mobility on the QoS. We saw that during inter-subnet handover, QoS guarantees may not be preserved. One of the major problems is the interworking of RSVP with IPv6. IPv6 proposes the use of temporary addresses, called care-of-address, in case of inter-subnet mobility. In RSVP, all the reservations are identified with the IP source and destination addresses. Moreover, the RSVP module is not aware of mobility so all the flow states and reservations are based on the MT's fixed address,

the home address.

We identified three different problems. The first problem was the routing problem for RSVP control messages. The second problem was the flow mismatch problem where the intermediate routers can not identify the packets belonging to an RSVP session when an MT obtains a new temporary address. The third problem was a timing problem where we want to reserve resources along the new path quickly when an MT moves to another network. If the reservation of resources in the new path can not be taken effect immediately, QoS traffic will be temporarily exchanged over the best effort channel. We proposed a complete minimal solution with some optional enhancements to resolve these problems. These solutions have also been proposed to the IETF as an Internet draft [FHNS98]. As a summary, we proposed several modifications in order to improve the inter-subnet handover:

- Old MRs can forward packets to the new location of the MT, reducing potential packet losses during inter-subnet handover.
- Mobile IP and RSVP may interact to minimize QoS degradation during handover.
- QoS can be reserved quickly within the wireless network even when the MT moves to another access network. This can be achieved by the previous MR or the MT informing the new MR about its QoS requirements.

Simulations may provide more insight on the performance of our solutions for the support of RSVP support in mobile IPv6 environments. The performance of the solutions will certainly depend on different parameters such as traffic characteristics, mobility patterns of the terminals and network topologies. It is a point that needs further investigation. Finally our contributions here are the participation at the problem identification and the proposal of solutions to resolve the problem of RSVP support in mobile IPv6 in a multicast scenario [FHNS98] [FHNS99].

Chapter 5

Multicast Management

Multicasting is the process of delivering a packet to several destinations using a single transmission [DDC97]. The advantage of multicast communication is its efficient savings in bandwidth and network resources since the sender can transmit the data with a single transmission to all receivers. Multicast applications are becoming more and more popular. Examples of such applications include audio and video conferencing, distributed games, and *Computer Supported Collaborative Work* (CSCW). The key idea of these systems is based on multicast data transmission. Due to these advantages, it is important that our proposed wireless network can support multicast communications. The important issues concerning multicast communications are:

- How to trace group members in a local network?
- How to deliver multicast traffic to the receivers?
- How to address a group at the network layer as well as at the link layer?
- What is the effect of mobility on multicast communication?

IP multicast standard provides a group management protocol to trace group members in a local network. However, further investigation is necessary in order to evaluate the suitability of this protocol in a wireless network. Group members can be all in a local network or distributed in different networks. Therefore, we have to provide not only a local mechanism to deliver multicast traffic to the senders inside a network but also a global mechanism to transmit the multicast traffic to other networks. IP multicast does not deal with the local delivery of multicast traffic. It just relies on the link layer mechanisms for the multicast traffic distribution in the local network. Multicast traffic transmission on the radio channel is performed by the APs. In order to optimize the bandwidth use, we have to take advantage of the native broadcasting capability of the radio interface to transmit multicast traffic in each cell. Furthermore, in a bandwidth-scarce environment like radio channels, multicast packets must only be forwarded to the APs with active members of the specified group. Multicast traffic delivery between different network domains is assured by multicast routing protocols. Multicast routing protocols are outside the scope of the design of a wireless IP network and therefore are not considered further.

In order to recognize the multicast traffic at the network and link layers, we need a group addressing scheme at these layers. At the network layer, a group can be identified thanks to the IP class D addresses. At the link layer, we need to provide a special addressing mechanism in order for the APs to identify the multicast traffic. Finally, we have to study the possible effects of mobility on multicast communication.

As a result, the wireless interface imposes specific requirements on the multicast scheme. Both the network layer and the radio link layer must be modified to meet these requirements. The radio link layer is in charge of multicast traffic delivery in the local network. It also provides a link layer addressing scheme for the groups. The network layer is responsible for routing multicast traffic to other networks. It also provides a way to trace group members in its local network.

This chapter presents the multicast management scheme used in our proposed wireless IP system. We first describe the IP multicast standard, specially its group management protocol. We then present wireless link issues requiring some modifications in the standard for an efficient multicast communication. We introduce a specific group management protocol designed for wireless networks. Performance evaluation shows that this protocol is suitable for wireless access networks. The effect of mobility on multicast communication is also considered in this chapter. Finally, we present the detailed description of our proposed multicast communication scheme in the wireless IP system.

5.1 IP Multicast

IP Multicast is a series of extensions made to IP in order to support multicast communication. A group of hosts is identified in IP multicast by an abstract class D IP address. A source can send data to a group address without knowing the group members. IP multicast standard uses the *Internet Group Management Protocol* (IGMP) [Dee89] in order to keep track of group members in a local network. For this purpose, a router, called *multicast router*, is required to be in charge of group management in its local network. A multicast router needs to know the presence of each group in its local network. A packet destined to a group will be transmitted in a local network if the specified group has at least one member in the local network. IP assumes that the underlying link layer distributes the packet to the members of the multicast group. Wide area multicasting is supported via multicast routing protocols. These protocols need to know the list of all networks having a member of a group in order to build a multicast tree for the group. IGMP provides the list of all groups present in its local network to multicast routing protocols.

IGMP is based on a *soft state* model. The multicast router periodically sends a *query* message to all hosts address. On reception of a *query* message, each host sends a *report* message for each group in which it participates. A *report* message for a group is sent to the group address so that every group member can hear it. On reception of the first *report* message for a group, other group members suppress their membership *report* for that group. The multicast router updates its group membership list after receiving each *report* message. If no *report* message is received for a group after several *query* messages, the router assumes that there is no group member in its local network and deletes the group from its list. As it

can be deduced, a multicast router only knows the presence of a group in its network. The identity of the members as well as the number of members in each group are not known due to the suppression mechanism of *report* messages.

In IGMP, a host wanting to join a group, sends an unsolicited *report* message for that group. Leaving a group does not require any explicit action. This introduces a *leave latency* between the time when a host, which is the last member of a group, really leaves the group and the time when the multicast router detects the situation and stops forwarding the traffic. IGMPv.2 [Fen97] attempts to decrease this latency by introducing two new messages: *leave* message and *group specific query* message. A host, wanting to unsubscribe itself from a group, sends a *leave* message if it is the last member of the group. On reception of a *leave* message, the router sends a *group specific query* message for that group in order to make sure that there is no other member of the group in its local network. *Group specific query* message acts like a *query* message but it is only destined to one group, the group for which the multicast router received a *leave* message. The advantage of *group specific query* message is that the time interval between two *group specific query* messages can be short, causing a shorter leave latency. In IGMPv1, the time interval between two *query* messages must be long enough to allow different groups to send their *report* messages.

5.2 Wireless Link Issues

The IGMP mechanism for group membership management is well adapted to classical LANs where bandwidth is not a scarce resource. The periodic transmission of *query* messages in IGMP causes not only the waste of bandwidth but also a high power consumption in the MTs since it prevents the MTs to go to sleep mode. A wireless LAN differs from a wireline LAN in many aspects. A wireless network is physically divided among different cells managed by different APs. An MT local to an AP can not receive the data from the other AP, although the two APs are located on the same IP subnet. The MTs located in the same cell can only hear the data coming from their AP. Hence, the multicast router must send a *query* message to all APs in its IP subnet in order to reach all MTs in the local network. On the other hand, the *report* messages sent by an MT may not be heard by all MTs immediately and a loop-back mechanism may be required for the multicast router to retransmit the message to the MTs in other APs.

Other issue is the problem of detecting the situation when a group has no more members. In IGMP, the *query* and the *group specific query* messages are not sent reliably. Therefore, the multicast router must repeat them several times before assuming the group absence in its local network. The number of times a *group specific query* or a *query* message is sent by the multicast router before assuming the group absence is called the *Robustness factor* R_f. The Robustness factor can be tuned by the network administrator according to the expected packet loss rate of the network. Choosing an optimal value for R_f is quite difficult due to the variable nature of error rate in wireless links. Packets may experience different error rates due to fading effects. On the other hand, link layers are normally equipped with error control mechanisms because of the high error rate of the radio medium. Therefore, it is possible that the IGMP packet goes under several retransmissions at the link layer before being accepted due to errors. This may cause the IGMP to decide that there is no member of the group in its

local subnet, while the link layer is trying to get the packet across the link.

Yet, the most serious problem comes from the leave latency. During the period of leave latency, the multicast router forwards the multicast traffic to the network while there is no receiver for the traffic. In IGMPv1, this period is quite long. In IGMPv2, this period is shorter than IGMPv1 but still exists. [Riz98] proposed a mechanism to decrease this leave latency based on prediction techniques. The multicast router maintains a history of the last *query's* outcome. On reception of a *leave* message, the router tries to predict the outcome based on the recorded history. It also sends a *query* message in order to make sure about the correctness of its prediction. In a wireless environment, we have a high error rate. Therefore, the probability to have a corrupted history is higher than the fixed links, especially in the case of fading when normally it takes some time before the channel returns to a better state. Furthermore, this scheme does not suppress the periodic transmission of *query* messages which causes the waste of bandwidth and a high power consumption in a wireless network.

[XP98] proposed an explicit join/leave mechanism to control group membership for point-to-point local links. A host sends an explicit *join* message when it wants to receive data from a group. It sends a *leave* message when abandoning a group. The existing *report* and *leave* messages in IGMPv2 can be used as explicit *join* and *leave* messages accordingly. The robustness of the protocol is assured by the router sending an acknowledgment when receiving a *join* or *leave* message. If a host does not receive an acknowledgment from the router after a time interval, it repeats its *join* or *leave* request. [XP98] proposed the use of this mechanism in point-to-point links in order to save bandwidth. Our performance evaluation shows that the use of join/leave mechanism leads to less overhead even if we use a shared medium (the multicast traffic is distributed to the receivers using a broadcast mechanism). This approach eliminates the leave latency and periodic transmission of *query* messages. It is well adapted to a wireless network because of its bandwidth savings and low power consumption in the MTs.

5.3 Wireless Group Management Protocol

The essential design criteria for a multicasting scheme in a wireless network is to avoid the waste of bandwidth and to use the broadcast nature of radio for multicast traffic delivery. In order to fulfill these design criteria, we propose a group membership protocol, that we call *Wireless Group Management Protocol* (WGMP), based on the explicit join/leave mechanism described in [XP98]. An MT sends a *join* or a *leave* message in order to subscribe or unsubscribe itself. These *join* and *leave* messages are confirmed by an *acknowledgment* message coming from the MR. An MT must retransmit its *join* or *leave* message if it does not receive an acknowledgment from the multicast router after a timeout. The timeout value can be quite short in comparison to *query* and *group specific query* interval timers since it must only account for the round trip time of the local network and the processing delay.

As stated before, IGMP requires the multicast router to maintain a list of groups present in its local network. This group presence list is not sufficient for an efficient multicast communication in a wireless network. In order to avoid the waste of bandwidth, multicast packets must be forwarded only to the APs with active members of the specified group. This necessi-

tates the multicast router to keep more information than the list of present groups in its local network. An MT can be anywhere in its local network. [AB96] proposed the idea of keeping a *host view* per multicast group. A host view of a group represents a set of cells in which every member of the group resides. Thus, instead of individually following each MT which belongs to a group in its local network, the group location is tracked.

In our system, the MR keeps a *Group Location Information* (GLI). Each group, that has a member in the local network, must have an entry called *location list* in the GLI. A location list contains a set of APs. An AP belongs to the location list of a group if at least one MT belonging to that group is located in its cell. Therefore, whenever there is traffic destined to a group, the MR consults the location list of the group in the GLI. It then forwards the traffic only to the APs of the group location list. The MR stops forwarding data to a group whenever the group location list is empty in the GLI. In this case, the MR must also prune itself from the corresponding multicast tree.

There are some situations where an MT loses its connection to its AP due to fading effects or a handover. If this MT is the only member of a group in its AP, a mechanism is needed to stop the transmission of multicast traffic on the AP. GLI by itself is not sufficient in this case since it only provides the location of group members in the access network. We propose the MR to maintain a *Group Membership Information* (GMI) which is the list of all MTs subscribed to a group in its local network. Normally in wireless networks, the link layer detects if a mobile terminal is no longer alive. The WGMP must be notified by the lower layers if an MT is off or if it is in another network domain. On reception of a notification by the lower layers, WGMP can identify the groups to which the MT has subscribed thanks to the GMI. It then makes the necessary modifications in the GLI and GMI.

An MT can move to another local network. When away from its network domain, the MT must be able to receive multicast data via its HA [Per98]. In this case the MT must tunnel its group membership messages to its HA and the HA must act as the multicast router of the MT, forwarding the multicast datagrams down the tunnel to the MT. The tunnel can be viewed as a point-to-point link with the HA at one end and only one receiver, the MT, at the other end. The HA may end up maintaining the list of subscribed groups for each MT outside its home network. In the WGMP, the MR maintains the list of subscribed groups for all MTs while in normal case, this list must only be maintained for MTs which are in a foreign network.

The GLI may be updated either because of terminal mobility or because of membership changes. Changes due to terminal mobility are discussed in the next section. Membership changes cause an update of the GLI in two cases:

- First join case and
- Last leave case.

The first join case corresponds to the situation when an MT sends a *join* demand for a group that has no member in its AP. In this case, the corresponding AP must be added to the GLI. The last leave case corresponds to the situation where an MT, which is the only member of a group in its AP, sends a *leave* message for the group. In this case, the corresponding AP

must be deleted from the GLI. The GMI however is updated with each membership changes. An MT sending a *join* message for a group will be added to the GMI of the group. It will be removed from the GMI when it requests to leave the group.

While the existence of the GMI is necessary to avoid unnecessary forwarding of the multicast traffic on the access network, the use of the GLI is optional. In fact, the concept of GLI was introduced in order to reduce the processing load of the router each time a multicast datagram arrives at the access network. Using the GLI, the multicast router can simply forward the datagram on the corresponding APs, otherwise it has to locate the group members before sending the data.

Finally, in order to support wide area multicasting, the WGMP must provide the list of groups present in its local network to the multicast routing protocols. This list can be easily generated from the GMI. A group is present in the local network if it has an entry in the GMI.

In WGMP, we maintain a *hard state* in the multicast router as opposed to the *soft state* in IGMP. Soft state requires several timers in order to refresh the data kept in the multicast router. In both IGMP versions, when a host receives a general *query* message, it sets delay timers for each of its subscribed groups. Therefore in each host, a timer is required per group. On the other hand, when a multicast router receives a *report* message, it adds the new group to its group presence list and sets a timer for the group. Hence, the multicast router also requires a membership timer per group. The use of complex timers in IGMP causes a high power consumption in MTs while the only requirement of WGMP is some additional memory in order to maintain extra information in the multicast router.

5.4 The Effect of Mobility on Multicasting

Let us first consider the intra-subnet case where an MT makes a handover to an AP under the same network domain. In this case, the IP layer is unaware of terminal mobility since the MT does not change its IP address. Host mobility will cause an update in the location list of a group either if an MT, which is the last member of a group in its AP, leaves its cell or if an MT enters a cell which has no member of its subscribed groups. If after a handover a group has no other member in the old AP, the MR must delete the old AP from the GLI. The MR must add the new AP to the GLI of those groups that had no members in the new AP prior to the MT's handover. Figure 5.1 shows an example of an intra-subnet handover. The GMI does not need to be updated since the MT is still a member of the group though in another cell.

The situation is more complicated in case of inter-subnet handover, where the MT moves to an AP outside its current network domain. Inter-subnet handover follows the IPv6 mobility functions as stated before. Here the important issue is the mechanism that allows the MT to send or to receive multicast traffic in a foreign network.

When entering a foreign network, the MT has two possibilities to receive multicast traffic. It can receive it via its HA. This requires the HA to be a multicast router in order to process all the *join* and *leave* messages coming from the MT. This approach leads to sub-optimal routing

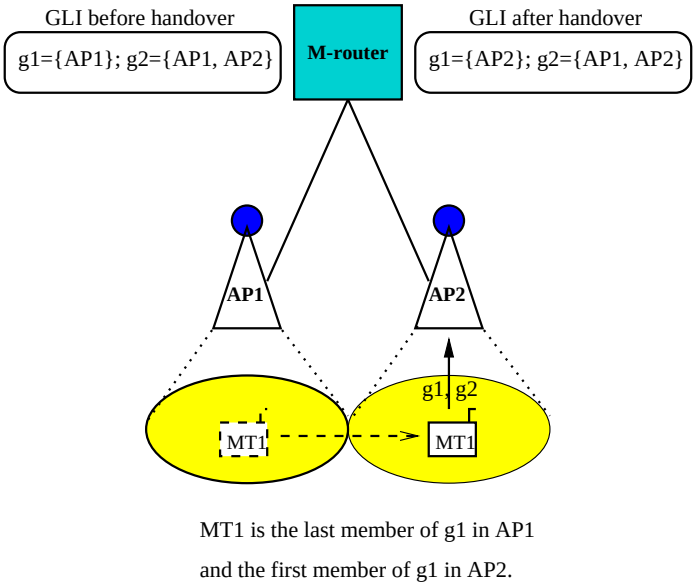


Figure 5.1: Intra-subnet handover

where all multicast packets must first be sent to the mobile terminal's HA and then tunneled to the foreign network. It is specially not desirable for the real-time applications due to the extra delay caused by the triangular routing. On the other hand, it ignores all the advantages of multicast transmission in terms of resource consumption due to the fact that a copy of multicast packet must be sent to each MT outside its home network. This solution requires the GLI to be updated accordingly in order to reflect the new location of the MT (foreign network). The GMI, however, does not need any update because the multicast traffic is still delivered to the MT by its home network.

The other solution is that the MT rejoins its multicast groups in its foreign network. The foreign network executes the group membership protocol and delivers the multicast traffic to the MT directly. This approach is preferable to the first one because of its optimal routing and efficient bandwidth consumption. However, the MT risks to lose packets during handover. This is due to the fact that when an MT, that belongs to a group, enters a foreign network with no member of that group, it can not receive the group traffic immediately. The local multicast router needs to graft a path to the multicast trees of the group with respect to all active sources of the group. This situation is depicted in Figure 5.2. The packet losses can be avoided by the previous router acting as an HA and forwarding the multicast traffic to the MT during the transient state of the handover. The MR must remove the MT from its GMI. The MR must also update its GLI in case the MT was the only member of its group in its cell. No assumptions are needed to be made for the group management protocol of the foreign network. In fact, if the foreign network deploys the IGMP protocol, it will not cause any interruption on the multicast communication of the MT. However if the foreign network deploys WGMP, the MT must be added to the GMI of the network. Moreover, the GLI must also be updated in the foreign network if the MT is the only member of the group in its cell.

In the same way, the MT can send a datagram to a multicast group in two ways in a foreign network. It can send it either via its HA using its home address or directly on the

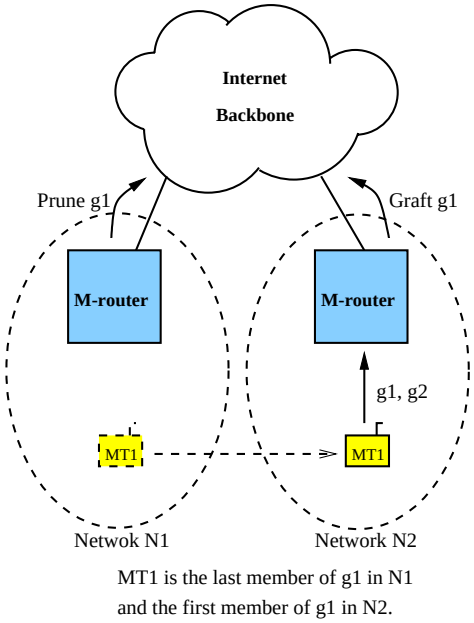


Figure 5.2: Inter-subnet handover

foreign network using its care-of-address. The first approach leads to sub-optimal routing and extra delay due to the fact that datagrams must be first forwarded to the HA. The second approach is optimal in bandwidth use and routing. The MT must use its care-of-address as the IP source address when sending datagrams directly from its foreign network since some multicast routing protocols use the IP source address in order to route a datagram. Note that the MT changes its care-of-address according to its location in the Internet. Therefore, a mechanism is needed for the recipients to know that although two multicast datagrams contain different source addresses, they originated from the same mobile sender which has moved across different networks. IPv6 introduced a new header field called home address options header. This header field is only processed at the destination points. We propose that the MT inserts a home address option containing its home address in its IPv6 datagram when sending multicast traffic in a foreign network. The receivers can then identify the sender by its home address.

5.5 Performance Evaluation

In this section, we compare WGMP with IGMPv1 and IGMPv2 in terms of protocol overhead. [XP98] compared the overhead of IGMPv1 and IGMPv2 with the join/leave approach but for a point-to-point link assuming no packet losses. Packet losses are quite common in wireless links and they may affect the performance of the protocols. We compare the overhead of WGMP with both versions of IGMP in two cases, once in the presence of packet loss and once without any packet loss. We take a shared link where a broadcast mechanism is used for multicast transmission in each cell. We use the default values specified in [Fen97] for different timers and variables of IGMPv1 and IGMPv2. These variables as well as their default values are shown in Table 5.1.

Query interval time T_q is the interval between two *query* messages sent by the multicast router. *Query response interval* T_r is the maximum allowed time for a receiver to send its *report* message in response to a *query* message. Using these default values and according to [Fen97], IGMPv1 requires $R_f T_q + T_r$ seconds to notice group absence in the worst case where the last member of a group leaves right after a *query* message. Note that the router repeats its *query* message R_f times before assuming that a group has no more members in its local network. *Query specific interval* T_{qs} is the time between two *group specific query* messages. It is also the maximum allowed time for a receiver to send a *report* message in response to a *group specific query* message. For IGMPv2, the leave latency is calculated as $(R_f + 1)T_{qs}$ [Fen97].

T_q	query interval time, default value 125 seconds
T_{qs}	query specific interval time, default value 1 second
R_f	robustness factor, default value 2
T_r	query response interval time, default value 10 seconds

Table 5.1: The definition of different parameters of the IGMP and their default values

The overhead of each approach consists of its control messages plus the data sent by the multicast router on its local network during the leave latency time. We do not account for the overhead of the general *query* and *report* messages since their costs are amortized over all the existing groups of the local network. In both IGMP versions, when a host wants to join a group, it sends two unsolicited *reports*. Therefore, the overhead of IGMPv1 is due to the two unsolicited *report* messages that each group member sends, plus the data sent by the router to the local network during leave latency. The overhead of IGMPv1 for a given group can be calculated as follows:

$$overhead_{(IGMPv1)} = 2LN + (R_f T_q + T_r)D \tag{5.1}$$

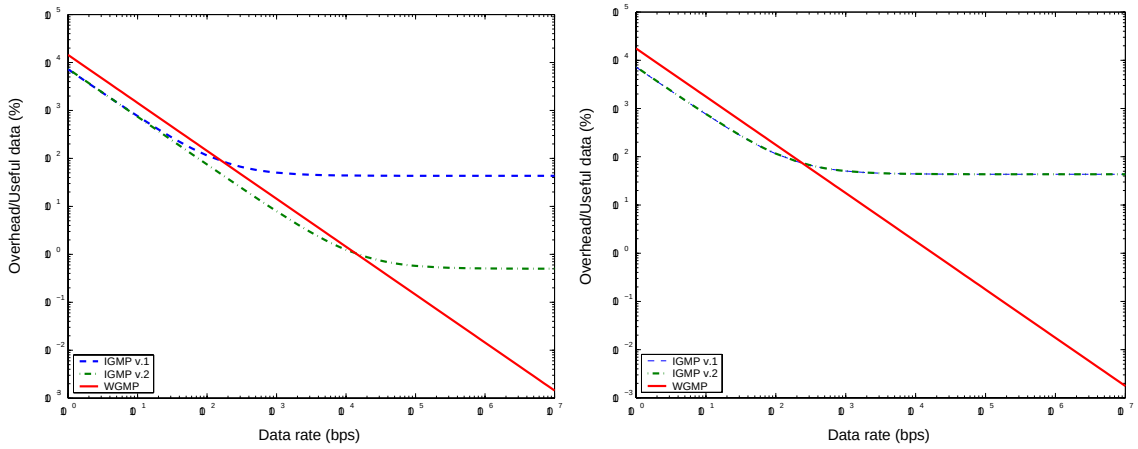
where L is the packet size, D is the data rate of the channel and N is the number of members in the group.

The overhead of IGMPv2 is due to two unsolicited *reports* of each group member, $2LN$, one *leave* message corresponding to the last member of the group in the network, L, R_f *group specific query* messages, $R_f L$, and the unused data sent during the leave latency $(R_f + 1)T_{qs}D$.

$$overhead_{(IGMPv2)} = 2LN + (R_f + 1)L + (R_f + 1)T_{qs}D \tag{5.2}$$

In WGMP, the overhead is due to the control messages since there is no leave latency if no packets are lost. The control messages of WGMP consist of a *join* message, a *leave* message and two *acknowledgments*.

$$overhead_{(WGMP)} = 4LN \tag{5.3}$$



(a) No packet loss

(b) Packet loss

Figure 5.3: The percentage of overhead over useful data as a function of data rate for different group management protocols, N=50

The first plot in Figure 5.3 depicts the overhead of each approach as a percentage of useful data as the data rate changes. No packets are lost in this case. We have fixed the membership duration at 10 minutes and the number of group members at 50 MTs. IGMP payload has a total size of 8 bytes [Dee89]. In IPv4, IP header fields take at least 20 bytes without any header options. For IPv6, the minimum length of header fields become 40 bytes. Thus, we get a total size of 28 and 48 bytes for the IGMP control packets at the IPv4 and IPv6 levels respectively. The WAND radio link layer uses a fixed packet size of 54 bytes containing 48 bytes of payload and 6 bytes of header. Therefore at the link layer, an IGMP control packet can be easily transmitted in a 54 bytes packet.

As it can be seen in Figure 5.3, with low data rates the overhead of control messages is quite high. As the data rate increases, the overhead of control messages become negligible and the effect of leave latency becomes dominant. In WGMP, the overhead decreases linearly as data rate increases. This is due to the fact that the overhead of WGMP is fixed (4 control messages for each group member). The first plot shows that the overhead of WGMP becomes less than the two versions of IGMP if data rate exceeds 10 Kbit/s. It justifies the use of WGMP in the our proposed wireless access network whose data rate is 20 Mbit/s.

The overhead of WGMP increases when a *leave* message corresponding to the last member of a group is lost. We consider this worst case for WGMP and we compare it with the same case in IGMPv2. Losing a message in IGMPv1 does not have any effect on its overhead since the messages are transmitted periodically anyhow. In IGMPv2, when a *leave* message is lost, the protocol behaves exactly like IGMPv1 since there is no retransmission for the *leave* message. The router notices the group absence when the group times out as in IGMPv1.

In WGMP, when a host sends a *leave* message, it waits for an acknowledgment from the router. If it does not receive the acknowledgment after a timeout T, it retransmits the *leave* message. We suppose that the packet losses are independent events. In this case,

the probability that an MT sends its *leave* message m times before the router can receive it correctly is $p^{m-1}(1-p)$ where p is the probability of packet loss. The average number of transmissions necessary for the correct reception of the *leave* message, $E[M_r]$, is:

$$E[M_r] = \frac{1}{1-p} \tag{5.4}$$

The overhead of WGMP in the worst case is calculated below. Note that we assumed only the *leave* message coming from the last member of the group is lost. The overhead of WGMP is due to the *join* message and its acknowledgment of the last member, $2L$, the *leave* message and its acknowledgment of the last member, $(E[M_r]+1)L$, the control messages of the other members, $4L(N-1)$, and the data sent on the network during the retransmission phase of the *leave* message, $(E[M_r]-1)TD$ with T representing the retransmission timer.

$$overhead_{(WGMP)} = 4L(N-1) + (3+E[M_r])L + (E[M_r]-1)TD \tag{5.5}$$

The second plot of Figure 5.3 shows the percentage of overhead over useful data when a *leave* message gets lost. The retransmission timer T is fixed at two times the round trip time of the local network. The number of group members is 50 MTs and the membership duration is again 10 minutes. The packet loss probability is 0.01. With these values, we observe that WGMP offers less overhead than the other two protocols for the data rates higher than 200 bps!

Next we are interested to see the effect of the number of members in a group on the overhead of each scheme. We fix the data rate at 1 Mbit/s and the membership duration at 10 minutes. As it can be seen in Figure 5.4, WGMP has less overhead than IGMPv2 for $N < 2000$ and has less overhead than the IGMPv1 for all the depicted values of N. This result also justifies the use of WGMP in our proposed wireless access network since we can rarely have so many members participating in one group in a wireless LAN. Even in a fixed LAN, we can hardly have so many members.

Figure 5.5 shows the percentage of overhead over the useful data as a function of membership duration. Membership duration has been varied from 1 minute to 100 minutes in order to see the effect of membership time dynamics on the behavior of each protocol. We have fixed the data rate at 1 Mbit/s and the number of members are $N = 50$. We can see that for short membership durations, the overhead of each protocol is the dominant factor of the curves. For long membership durations, this overhead is amortized over the total amount of useful data sent on the network and that is why the curves are stabilized after a certain membership duration.

5.6 Multicast Management Scheme

In our proposed wireless system, the MR is in charge of group membership management in its local network. The WGMP is executed in the MR as well as in the MT. The AP has no

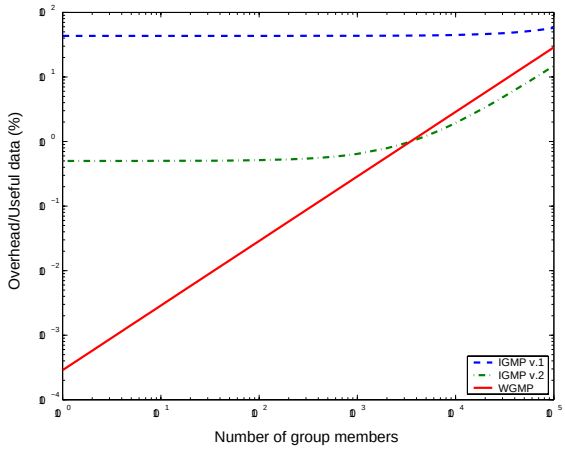


Figure 5.4: Overhead as a function of number of receivers, no packet loss, D=1 Mbit/s

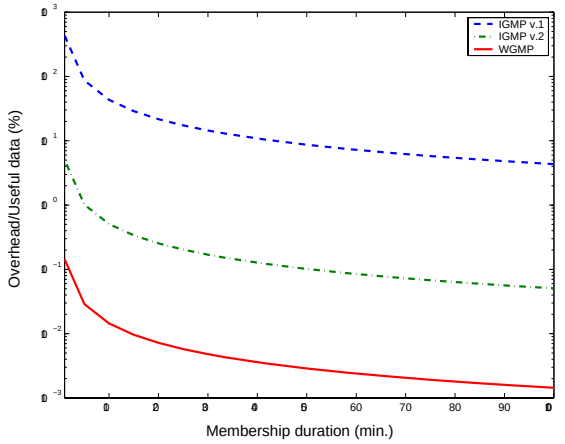


Figure 5.5: Overhead as a function of membership duration, no packet loss, D=1 Mbit/s, N=50

control over group membership management. It only forwards the multicast control packets between the MR and the MTs. The MR processes all the *join* and *leave* messages coming from MTs. It updates the GLI and the GMI if required.

The radio sub-system of WAND provides a link layer addressing scheme. Each MT is identified by a link layer address in the AP. This address is given by the AP to the MT and is unique in the range of a single AP. The link layer addressing scheme can be similarly used to identify a multicast group in the AP. The radio sub-system also provides some facilities for broadcast channels. These facilities together with link layer addressing of multicast groups can be used for efficient transmission of multicast traffic in each cell. The mechanism is as follows. The MR orders an AP to allocate a link layer address to a group whenever there is a join demand for a group that does not have any member in that AP. The AP uses this address to identify the group as long as the group has at least one member in its cell. This address is communicated by the AP to its local group members. The group members, in turn, memorize this address for further use. The AP delivers traffic coming for a group using the link layer address of the group. Therefore, only the MTs which are aware of this address

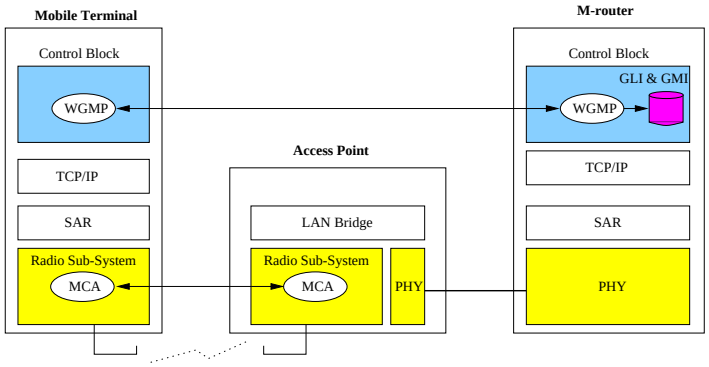


Figure 5.6: Architecture of the multicasting scheme

will receive the multicast traffic. Other MTs simply discard the traffic since it has not been destined to them. In consequence, multicast communication requires some modifications in the radio link layer. The *MultiCast Agents* (MCA) added to the radio sub-system of the MT and the AP are responsible for the multicast functionality in the link layer, as it is shown in the Figure 5.6.

The notion of IP and radio flow have already been introduced in Chapter 3. In order to facilitate the multicast communication, we propose the definition of a default flow identifier at the network level and a default radio flow identifier at the radio link layer for multicast. At the network layer, the combination of flow identifier, the source and destination IP addresses allows us to separate different traffic coming from different sources. At the link layer, the combination of radio flow identifier, AP link layer address and MT link layer address allows us to differentiate multicast traffic. Note that all multicast traffic are sent as best effort. No special QoS mechanisms have been provided in our system for multicast communication.

In case of handover, multicast communications are essentially handled in the same way as it is described in Section 5.4. The MR updates the GLI and GMI if necessary. In addition, if a group has no other member in the old AP, its link layer address as well as its active flows must be released. On the other hand, the new AP must allocate new link layer addresses to those groups that had no members in its cell up to now. In the inter-subnet handover case, the MT must rejoin its multicast groups in its foreign network. The foreign network delivers multicast traffic to the MT directly. The MT sends multicast traffic directly in its foreign network using its care-of-address as its source IP address and its home address in the home address option header field.

The overall mechanism for joining or leaving a group is summarized below. An MT subscribes itself to a group by sending a *join* message to the MR. A join process requires the following steps:

- An MT wanting to join a group g sends a *join* message to the MR using the multicast address of g.
- The MR determines if the MT, that issued the *join* message, is the first member of g in its AP. If this is the case, the MR must ask the corresponding AP to allocate a link layer address for g. The AP must also create the default radio flow.

- The AP then communicates the link layer address of *g* to the MT. The MT adds the link layer address of *g* to its group list. It also creates the default radio flow for the best effort channel with the same identifier allocated by the AP.
- If the MT is the first member of *g* in its AP, the MR adds the AP link layer address to the GLI. The MR adds also the MT to the GMI.

An MT unsubscribes itself from a group by sending a *leave* message to the MR. The following steps must be done when a MT wants to leave a group:

- An MT wanting to leave a group *g* sends a *leave* message to the MR with the multicast address of *g*.
- The MT deletes *g* from its group list. It also removes the radio flow.
- The MR determines if the MT is the last member of *g* in its AP. If this is the case, it asks the corresponding AP to release the link layer address allocated to *g* and to remove its radio flow identifier.
- If the MT is the last member of *g* in its AP, the MR deletes the AP link layer address from the GLI. It also removes the MT from the GMI.

5.7 Conclusion

This chapter presented a complete framework for multicast communication in the IP access system. Some of the problems of IP multicast standard in the presence of wireless links were discussed. A new group management protocol was proposed to resolve these problems. As the performance evaluation showed, the protocol has less overhead than its counterparts in IP multicast in most cases. In order to avoid sending a separate copy of each multicast packet to every recipient of a group, we deployed the broadcast nature of radio to transmit multicast traffic. At the link layer, we proposed the use of a link layer addressing scheme for multicast groups in APs. The link layer addressing allows the APs to identify the multicast traffic. Furthermore, the original WAND radio has already a link layer addressing mechanism which facilitates the deployment of our proposition.

The area of multicast communication is an active research area. There are lots of works that have been done on different aspects of multicast communications but all these works have supposed a fixed Internet environment. To our knowledge, not much research has been done for multicast communication in wireless networks. Wireless networks can benefit from multicast transmissions due to their native broadcast mechanism. The reliability mechanism for multicast communication in wireless networks is an interesting issue which requires further investigation. In the current design of wireless IP system, all multicast traffic are sent as best effort and no QoS has been provided to multicast traffic. We address this issue in Chapter 6. Finally, our contributions in this chapter are the problem identification, proposal of a new group management protocol, the performance evaluation of our proposed protocol as well as its counterparts in IP multicast and the proposal of solutions for multicast communication in case of terminal mobility [NB98] [NB99].

Chapter 6

The Use of FEC for QoS Control

As we saw in previous chapters, all the traffic in the access network are grouped into three different categories at the radio link layer. Each category corresponds to a radio QoS class. The first QoS class corresponds to real-time traffics that have strict delay requirements but can tolerate a high loss rate. An example of such traffics is voice. The second radio QoS class corresponds to traffics having medium delay constraints such as video. The last radio QoS class is appropriate for traffics that can tolerate high delays but are sensible to high loss rates. Each QoS class has its own reliability mechanism which is a function of its QoS requirements. The objective of the reliability mechanism is to control the packet dropping rate at the link layer.

In Chapter 3, we proposed the use of an ARQ/FEC scheme. The number of retransmissions was limited according to the delay requirements of each QoS class. The use of a fixed FEC was recommended for all classes but no information was provided on the coding scheme. [MLLV98] [MB99] [Mei98] evaluated and optimized the ARQ mechanism used in the WAND radio sub-system but to our knowledge, no work has been done on the use of FEC. Due to the advantages of FEC in some situations and for certain applications, it may be worthy to investigate the use of coding.

We also saw that in the framework of WAND, all multicast traffics are sent as best effort without any error recovery mechanism. Given that multicast applications are becoming more and more popular and that there are a lot of real-time multicast applications, there is also a need to provide QoS to multicast applications. If we consider multicast communication as a general communication mode where the traffic is sent to a set of receivers, unicast and broadcast communications can be viewed as special cases where the traffic is only sent to one receiver or to all receivers respectively. Having a framework for QoS provisioning in the case of multicast communications means that the same general rule can be applied to any communication mode. Note that we suppose a single QoS per multicast session. It means that all members of a given group are supposed to have the same QoS requirements. Other approaches like layered multicast [MJV96] may be used in the case of receivers with different QoS needs but this is not the subject of this study.

Several researchers have studied the error control protocols for radio channels combining coding with a retransmission scheme but they assumed a one-to-one or unicast communication mode [LY82] [WL83] [Kal90] [KH90] [GGB96]. On the other hand, the researchers in

the networking area have done several studies on error recovery mechanisms for multicast communication [FJL+97] [PSLB97] [NBT98] [PSA96] [GBS94]. A survey of reliable multicast protocols can be found in [LGLA98]. Most of the reliable multicast protocols propose the use of ARQ [FJL+97] [PSLB97]. However, the use of simple ARQ for reliable multicast transmission toward a large group may cause a high retransmission rate at the sender even if each receiver has a low error rate. The use of FEC in this case can reduce the retransmission rate tremendously [NBT98] [Hui96]. All of these researchers have considered an end-to-end error recovery in a fixed Internet environment. In fixed Internet, packets are most likely dropped due to congestion while in wireless, the unreliability of media is the major factor causing packet losses. End-to-end error recovery mechanisms do not necessarily work well in the presence of wireless links and different kinds of mechanisms are required to guarantee reliability at the traversed wireless links. These are our basic motivations for the study of error recovery mechanisms in the case of general multicast communication in wireless environments.

This chapter focuses on the use of FEC as a QoS control mechanism. Throughout this chapter, we use *Reed-Solomon Erasure* (RSE) codes because of their appropriate characteristics in terms of powerful coding and implementation simplicity. We take several QoS metrics in order to analyze the effect of coding. Two different error models have been taken, a *Binary Symmetric Channel* (BSC) model where error events are independent and a *Gilbert-Elliot* (GE) model where errors are correlated.

We start by some background information about coding and Reed-Solomon Erasure codes. Then, we specify our QoS metrics used in this chapter. Then, we evaluate the performance of coding in terms of our specified QoS metrics in BSC and GE models. For each error model, we present numerical results that compare the performance of an ARQ scheme with a hybrid ARQ/FEC scheme.

6.1 Coding Aspects

Coding consists of adding redundant information to data in order to allow the receiver to recover the original data even in the presence of transmission errors. Basically, a code transforms a *data block* of k symbols $d = (d_{k-1}, d_{k-2}, ..., d_0)$ into a *coded block* of n symbols $C = (c_{n-1}, c_{n-2}, ..., c_0)$. In a system that uses FEC for error control, the sender and the receiver use a mutually agreed code to protect the data. If a coded block can be divided into the data part and the redundancy part, then the code is said to be a *systematic code*. A systematic code generates a coded block consisting of an unaltered copy of the data block followed by the $h = n - k$ redundant symbols. The advantage of a systematic code is that in case a receiver receives the data block correctly, no decoding is necessary. *Redundancy* level of a coding scheme is defined as the ratio of h/k and it represents the amount of redundant information added to the original information.

There are two different types of codes in common use today, *block codes* and *convolutional codes*. A block code transforms each data block independently into a coded block of n symbols. In a convolutional code, however, each coded block depends not only on the corresponding data block but also on the m previous data blocks. The parameter m is some-

times called *memory order*. Convolutional codes work on a continuous bit stream generating n encoded bits every k data bits.

Convolutional codes are not effective in the presence of error bursts since each bit error in the received coded block impacts the decoding of m adjacent bits depending on the memory order of the code. Normally, convolutional codes are combined with *interleaving* in order to be effective against the bursty errors. Interleaving involves rearranging symbols from two or more coded blocks before transmission on the channel. Interleaving increases delay since all interleaved symbols must be received before decoding, making it feasible only for small data blocks. Thus in the following, we focus on block codes rather than convolutional codes due to the extra delay imposed by interleaving. Note that in order to mitigate the effect of burst errors that often characterize wireless channels, we can also use a large data block size instead of an interleaving scheme.

Forward error correction can be done at many levels. We distinguish three possible levels as in [APP+95] [LMZG97]:

- **Bit level FEC**: In a bit level FEC, a bit is considered as a symbol. Bit level FEC is basically implemented at the physical layer of almost all wireless networks. It is typically done by means of a *Digital Signal Processor* (DSP) chip or a specific *Integrated Circuit* (IC). It is designed to correct bit errors as its name indicates.

- **Byte level FEC**: This is done on a per-packet basis. Traditionally, each packet carries a *Cyclic Redundancy Check* (CRC) field for error detection. With the advent of more powerful processing abilities, the use of this field for error correction is also possible.

- **Packet level FEC**: Packet level FEC takes a packet as its coding symbol and generates h redundant packets from k original ones for error correction. In case of packet losses, these redundant packets together with the correctly received packets can recover the lost packets without any need for retransmission. Packet level FEC is based on erasure coding. In coding theory, an error is defined as a corrupted symbol in an unknown position while an erasure is a corrupted symbol in a known position. The error correcting capability of a code can be increased if the decoder can exploit the erasure information. Packet level FEC can easily retrieve the erasure information since the location of losses can be detected thanks to packet sequence numbers.

Bit level FEC is normally employed at the physical layer of almost all the wireless and mobile networks. Byte level FEC can correct mostly random errors in a packet but it is not efficient in the presence of long error bursts. Bit and byte level FECs can be used together with an interleaving scheme in order to cope with long fades. Packet level FEC, however, is quite efficient when dealing with long fades because of its large data block size. It is also a good choice at the link layer for systems that automatically drop the corrupted packets. Packet level FEC is also interesting in the context of multicast communication since a single redundant packet can recover the loss of different information packets at different receivers [NBT98] [Hui96].

[ES98] made a trace-based evaluation of the error characteristics of an indoor environment. They observed that the changes in the error environment over time are quite slow since

they are normally caused by human actions such as mobility. These slow changes will in turn cause long periods of similar error behavior. Simulations in [APP+95] also showed that indoor channels can benefit from packet level FEC due to the long periods of fading observed in such an environment. In the following, we concentrate on packet level FEC due to its capability to cope with bursty errors and its usefulness in case of multicast communication.

Block codes used in practical applications today belong to the class of *linear cyclic codes* since they lend themselves to easier implementations. A code is refered to as linear if the sum of two coded blocks is also a coded block. A linear code is called a *cyclic code* if every cyclic shift of a coded block is also a valid coded block. *Reed-Solomon* (RS) codes [RS60] are an example of linear cyclic codes. Reed-Solomon codes are in general non-binary. Each Reed-Solomon symbol is actually a group of bits. Just one bit error anywhere in a given symbol spoils the whole symbol. That's why Reed-Solomon codes are often called *burst error correcting codes* since they are more powerful if the bit errors are concentrated into as few symbols as possible.

Packet level FEC is also possible with Reed-Solomon codes when a packet is taken as a coding symbol. In this case, we refer to the Reed-Solomon code as Reed-Solomon Erasure (RSE) code. The advantage of using a Reed-Solomon code is that there are several implementations of this code at the public domain which facilitates the performance evaluation of the code. There may be other powerful erasure codes giving better results but our emphasis here is on the utilization of coding for QoS control rather than the type of code used.

6.1.1 Reed-Solomon Erasure Codes

A Reed-Solomon erasure code is a Reed-Solomon code with symbols defined over the *Galois Field* $GF(2^m)$, designed to correct only erasures. It is represented as $RSE(n,k)$ and it has a symbol size of m bits. A Reed-Solomon erasure code has the capacity to correct h erasures with only h redundant symbols. This characteristic makes this kind of code particularly powerful to cope with transmission packet losses. The parameters of such a code are:

$$\begin{array}{ll} \text{Number of symbols in a coded block:} & n = 2^m - 1, \\ \text{Number of redundant symbols:} & h = n - k, \end{array}$$

A coded block of n symbols, $C(c_{n-1}, c_{n-2}, ..., c_0)$, generated by any linear block code can be mathematically represented by a polynomial of degree n with the coefficients being the elements in the $GF(2^m)$.

$$C(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + ... + c_0$$

Using a systematic code, we can represent $C(x)$ as

$$C(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + ... + c_k x^k + d_{k-1}x^{k-1} + ... + d_0$$

where $c_{n-1}, c_{n-2}, ..., c_k$ are the h redundant symbols and $d_{k-1}, ..., d_0$ are the data symbols.

[LC83] proves that every cyclic code, $C(x)$, can be expressed as follows:

$$C(x) = d(x)g(x)$$

where $d(x)$ is the polynomial representation of the data block with $d_{k-1}, ..., d_0$ as its coefficients, and $g(x)$ is the *generator polynomial* of the code. The degree of $g(x)$ is equal to the number of redundant symbols. The generator matrix has the form:

$$g(x) = (x - \alpha)(x - \alpha^2)...(x - \alpha^h)$$

where α is the primitive element in $GF(2^m)$. Clearly, we have:

$$C(x) = 0 \qquad \text{for} \qquad x = \alpha, x = \alpha^2, ..., x = \alpha^h$$

Therefore, we can derive h independent equations giving us the h unknowns $c_{n-1}, c_{n-2}, ..., c_k$.

In the sender side, the RSE encoder takes k data packets and generates h redundant packets, as described above, to form a coded block of $n = k + h$ packets. If the receiver gets at least k packets out of the $k+h$ transmitted packets correctly, it can reconstruct the original data. Note that the loss unit is a packet and a packet payload is considered as a symbol.

6.1.2 Implementation Issues

McAuley proposed a hardware architecture for RSE codes in [McA90] using a symbol size $m = 8$ and $m = 32$. Rizzo proposed a software implementation of RSE codes in [Riz97] with a symbol size in the range of $m = 4, ..., 16$. RSE coders with large symbol size are difficult to implement. Normally, the packet size is in the order of hundreds or thousands of bits. In this case, we need to consider a packet as l symbols of m bits and the coding can be implemented using l parallel RSE coders, each operating on a symbol size of m bits.

Since the number of elements of the $GF(2^m)$ with a symbol size of m is limited to 2^m, it is important to choose an RSE code with $n < 2^m$. If we take $m = 8$, we will have a maximum block length $n = 255$ which is sufficient in our case.

In the following, we use the software RSE coder developed by Rizzo with a symbol size $m = 8$. The encoding and decoding speeds of this software coder have been tested in various platforms from high speed workstations to small portable systems [Riz97] [Rub98] and have been shown to be in the order of Mega Bytes per second. One important observation is that for all the tested platforms the encoding and decoding speeds, c_e and c_d respectively, remain approximately constant over a wide range of k and h with c_d slightly smaller than c_e due to additional overheads in decoding. As a result, we consider them as constants during our performance evaluations.

In order to have variable error correcting capabilities, we are interested to modify the coding parameters k and h of an RSE code. This is feasible by using *shortening* and *puncturing* techniques [LC83]. Shortening consists of adding a certain number of information symbols equal to zero to the original information in the encoding phase. Let's consider a Reed Solomon erasure code of $RSE(n, k)$. We can generate a set of shortened code $RSE(n-b, k-b)$ with $1 \leq b \leq k-1$ and an error correcting capability, h', equal to h. These shortened codes have their b high order information symbols equal to zero. Code puncturing involves not transmitting (deleting) certain redundant symbols. Puncturing allows a coder to change its number of redundant packets h while shortening allows it to change its number of data packets k. The shortened and punctured codes can use the same encoder/decoder pair

as their original code.

6.2 QoS Metrics

Several aspects must be taken into account when using a coding scheme as a QoS control mechanism. The first aspect to consider is the effect of coding on bandwidth. In other words, we have to evaluate how much overhead the coding scheme adds to the original scheme. We take *efficiency* as a measure of the used bandwidth and we define it as the inverse of the average number of transmissions required by all receivers to receive a packet correctly.

The second issue is to evaluate the loss probability before and after coding. We define *packet loss rate* as the probability that at least one receiver can not receive a packet correctly after the first transmission. This metric allows us to observe the decrease of loss rate due to the utilization of FEC. It gives us a precise measure of the effectiveness of a code in correcting errors.

The last issue is the effect of coding on delay. The transmission delay of a packet is composed of several components. At the sender side, the delay is affected by the processing, queuing and encoding time. At the network side, we have to account for the transmission and propagation delays. At the receiver side, there are also processing, queuing and decoding delays that have to be considered. We ignore the effect of the queuing delay since this delay can be influenced by other flows of data and it depends on the congestion state of the network and also on the scheduling algorithm used. Here our objective is to see the effect of coding on the average delay of a packet. In this sense, we are interested in comparing the average delay of a packet in an ARQ scenario with an ARQ/FEC scenario. Therefore, we define *packet delay* as the time spanning from the beginning of the transmission of a packet until it has been successfully received by all receivers.

Throughout this chapter, we suppose that the loss events at different receivers are independent. We assume that all bit errors in a received packet are detected thanks to its CRC field and no control messages are lost. In case of multicast, the traffic is transmitted to all receivers using the broadcast mechanism of the radio rather than sending a separate copy for each receiver.

6.3 Performance Evaluation of FEC in a BSC model

6.3.1 Binary Symmetric Channel Model

The Binary Symmetric Channel is an independent error model where every transmitted bit has exactly the same error probability as the other bits. The error process is a geometric process with the parameter e. The probability that a bit is transmitted erroneously is e and the probability that a bit is transmitted correctly is $1 - e$.

6.3.2 The Effect of FEC on Efficiency

Let us consider first the ARQ scenario where a sender multicasts data to R receivers. The sender retransmits the original packet if there is at least one receiver that has not received the packet correctly. In [TKP97], an expression is derived for average number of packet transmissions in a multicast group. We define M_r as the number of transmissions required for a correct reception of a packet by a receiver r and M as the number of transmissions required for a correct reception of a packet by all receivers. The average number of transmissions, $E[M]$, as well as the efficiency of the scheme, Eff, can be calculated as follows:

$$P(M_r \leq m) = 1 - p^m \tag{6.1}$$

$$P(M \leq m) = (1 - p^m)^R \tag{6.2}$$

$$E[M] = \sum_{m=1}^{\infty} m P(M = m) = \sum_{m=1}^{\infty} P(M \geq m) \tag{6.3}$$

$$Eff = \frac{1}{E[M]} = \frac{1}{\sum_{m=1}^{\infty} [1 - (1 - p^{(m-1)})^R]} \tag{6.4}$$

where p is the packet loss rate of each receiver. A packet will be lost if it has at least one bit in error. In a BSC model, p is calculated as:

$$p = 1 - (1 - e)^L \tag{6.5}$$

with e the bit error rate of the wireless link and L the packet length.

Now, we consider the ARQ/FEC scenario where the sender uses an RSE coding scheme $RSE(n, k)$. In this case, the sender sends k original packets followed by h redundant ones. Each receiver can recover from losses if it receives correctly k packets out of the $k + h$ transmitted packets, otherwise it asks for a retransmission. [NBT98] made a complete analysis of the average number of packet transmissions in this case. The perceived packet loss rate by each receiver, q, and the efficiency of the scheme are calculated as follows:

$$q = p\left(1 - \sum_{j=0}^{n-k-1} \binom{n-1}{j} p^j (1-p)^{n-j-1}\right) \tag{6.6}$$

$$Eff = \frac{1}{E[M]} = \frac{k}{n} \frac{1}{\sum_{m=1}^{\infty} [1 - (1 - q^{(m-1)})^R]} \tag{6.7}$$

We take a large set of RSE codes with different values of k and h for our analysis. The packet length L is 54 bytes as in HIPERLAN II and WAND. Figure 6.1 shows the efficiency as a function of bit error rate in a group of 1000 wireless receivers for the described scenarios. In the first plot, we compare the efficiency of the ARQ scheme with ARQ/FEC schemes using three different codes. These codes have a different number of redundant packets h but the number of their data packets k is the same. In the second plot, we take three codes with different data block size k but with the same number of redundant packets. The efficiency as a function of number of wireless receivers for a bit error rate of 10^{-4} is depicted in Figure 6.2. From these figures, the advantages of FEC are evident.

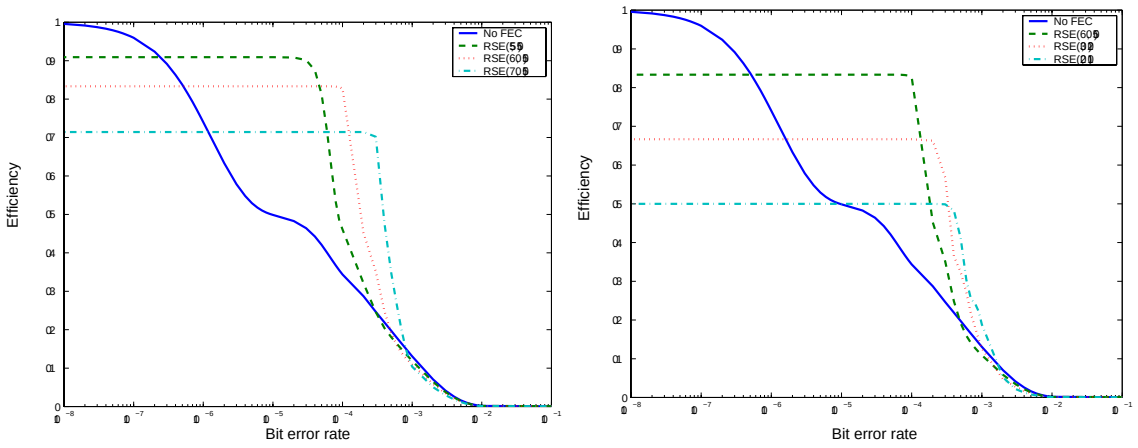


Figure 6.1: Efficiency as a function of bit error rate, R=1000

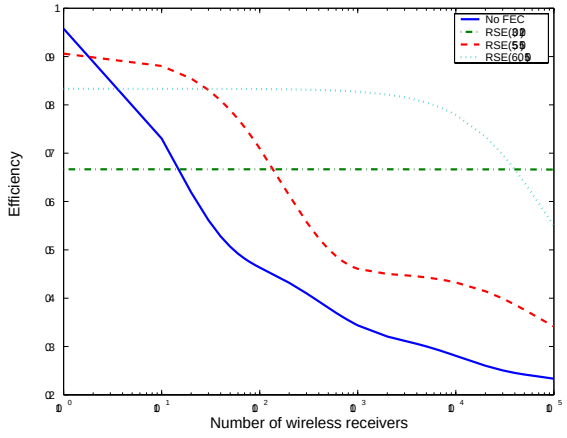


Figure 6.2: Efficiency as a function of number of wireless receivers, $e = 10^{-4}$

Based on these results, we can make the following conclusions:

- For low bit error rates, the best efficiency can be obtained by choosing a code with a high number of data packets, k, and a low number of redundant packets, h. If the bit error rate goes high, the number of redundant packets must be increased. Theoretically, if we assume to have an infinite number of redundant packets, we have to

increase h as a function of the channel bit error rate. Nevertheless, in reality we have only a limited number of redundant packets. In this case, if the maximum number of available redundant packets is not sufficient for an acceptable efficiency, we must decrease the number of data packets k while keeping the number of redundant packets at its maximum. As an example, if we look at the second plot in Figure 6.1, we observe that $RSE(60, 50)$ works well up to a bit error rate of 10^{-4} while $RSE(20, 10)$ keeps its efficiency at a constant rate up to 4×10^{-4}.

- The number of receivers has a big impact on the efficiency if only an ARQ scheme is used. The efficiency decreases sharply if the number of receivers increases significantly. The use of FEC reduces the impact of number of receivers on efficiency but its redundancy level must be chosen carefully. From Figure 6.2, we can observe that the $RSE(30, 20)$ maintains a constant efficiency for different number of receivers while the efficiency of the $RSE(60, 50)$ starts decreasing after 5000 receivers for a bit error rate of 10^{-4}.

- There is not one best code. Depending on the bit error rate and the number of receivers, the efficiency of a code varies. Therefore, we can only designate one best code for a range of bit error rates and number of receivers. If either the bit error rate or the number of receivers changes, the most efficient code also changes.

- For very high bit error rates, even a coding scheme can not help. A code is operational up to a certain bit error rate. Our set of codes are operational up to a bit error rate of 10^{-2}.

6.3.3 The Effect of FEC on Packet Loss Rate

The packet loss rate, PLR, of the ARQ scenario can be determined as follows:

$$PLR = 1 - (1 - p)^R \tag{6.8}$$

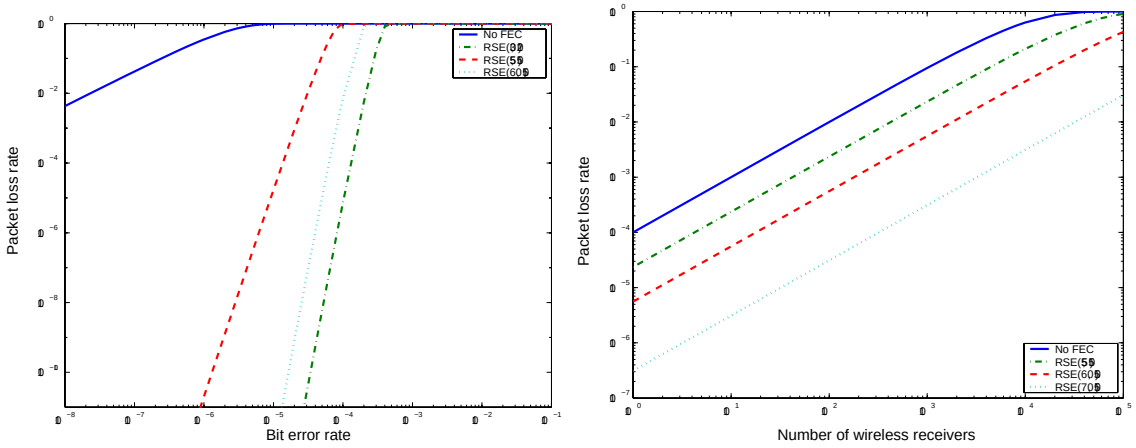
where p is the packet loss rate of each receiver found from equation (6.5). In presence of FEC, the PLR can be calculated as below using q from equation (6.6).

$$PLR = 1 - (1 - q)^R \tag{6.9}$$

In order to better understand the effect of coding on PLR, we compare the behavior of ARQ protocol with three ARQ/FEC protocols each using different RSE codes. The first plot in Figure 6.3 shows the packet loss rates of these schemes as a function of the channel bit error rate for a total of 1000 wireless receivers and the second plot shows the packet loss rate of the same schemes as a function of number of receivers for a bit error rate of 10^{-4}.

From these figures, we can observe that:

- An RSE code can guarantee a packet loss rate up to a certain bit error rate and up to a limited number of receivers.



(a) $R = 1000$ (b) $e = 10^{-4}$

Figure 6.3: Packet loss rate as a function of bit error rate and number of wireless receivers

- The more we increase the redundancy level of a code, the more a code is resistant to the increase in the channel bit error rate. For example, the $RSE(30, 20)$ can guarantee a packet loss rate lower than 10^{-10} up to a bit error rate of 4×10^{-5} while $RSE(55, 50)$ can guarantee the same packet loss rate up to a bit error rate of approximately 10^{-6}. The increase of number of receivers also increases the PLR of each scheme. However, a code with a higher level of redundancy is more resistant to the increase in the number of receivers as it can be seen in Figure 6.3.

- A coding scheme that can guarantee a low packet loss rate is not necessarily the best code in terms of bandwidth use. For example, the $RSE(30, 20)$ maintains a packet loss rate lower than 10^{-10} up to a bit error rate of 4×10^{-5} while in terms of efficiency, it is not the best code for this range of bit error rates as it can be seen in Figure 6.1. Choosing a code is therefore a trade-off between the guaranteed packet loss rate and the efficiency.

6.3.4 The Effect of FEC on Delay

We take two different error control protocols. In the first protocol, P1, we use an ARQ mechanism. The second protocol, P2, uses an ARQ/FEC scheme with RSE codes. In order to have a fair comparison of P1 and P2, we consider that both protocols send k data packets before waiting for a feedback. Protocol P1 proceeds as follows:

- At the sender side

  1. Send k data packets.
  2. Send a poll for feedback to indicate the end of the transmission and start a timer.

3. If no NAK is received after the time out, then no packet is lost. Proceed to the transmission of the next k new packets.
4. If there are R received NAKs or if there is a timeout but there are less than R NAKs, include the lost packets as well as new packets in the next transmission.
5. Go to the first step.

- At the receiver side

1. Buffer the received packets.
2. If k packets are received, send them to the higher layer.
3. If a poll is received but there is less than k packets in the buffer, send a NAK to the sender indicating the sequence numbers of the lost packets. Send the correctly received packets to the higher layer.
4. Go to the first step.

Protocol P2 uses a hybrid ARQ/FEC scheme. We assume that the receiver can distinguish a data packet from a redundant packet thanks to the header information of each packet. If the receiver receives all the k data packets correctly, no decoding is necessary. Protocol P2 performs as follows:

- At the sender side

1. Send k data packets followed by h redundant ones.
2. Send a poll for feedback to indicate the end of the transmission and start a timer.
3. If no NAK is received after the time out, then no packet is lost. Proceed to the transmission of the next coded block composing of k new packets and h redundant packets.
4. If there are R received NAKs or if there is a timeout with less than R NAKs, then include the lost packets as well as new packets followed by h redundant ones in the next transmission.
5. Go to the first step.

- At the receiver side

1. Buffer the data as well as redundant packets.
2. If all the k data packets are received correctly, send them to the higher layer.
3. If there are lost data packets but with enough redundant packets (at least k packets out of the $k + h$ transmitted packets are correctly received), recover the lost packets by decoding and send the k data packets to the higher layer.
4. If there are lost data packets but there are not enough redundant packets to recover (less than k packets are correctly received), generate a NAK including the sequence numbers of the lost packets. Send the correctly received data packets to the higher layer.

5. Go to the first step.

Note that both protocols are NAK based and that we send only one NAK for a block of k packets. The receivers send a NAK message if they have not received a packet correctly. If a receiver has received all the packets correctly, it does not send any feedback. It is clear that in this case the sender needs a timer mechanism in order to proceed the transmission if it receives no NAK message from receivers.

We have considered a non-continuous mode protocol where the sender sends a POLL message in order to inform the receivers about the end of the transmission. The sender then starts a timer and stops sending packets until it receives a NAK from each receiver or until it makes a timeout. This is because in case of multicast communication, the reception of one NAK can not trigger the transmission of the next block since it may be other receivers that have lost a packet in the transmitted block but they have not yet sent their feedbacks. Hence, the sender can not start the transmission of the next block. However if it has received either no NAK or less than R NAKs after the timeout, it means that other receivers have correctly received their packets and thus the sender can proceed to the transmission of the next block. It is clear that the timer value must be large enough in order to allow all receivers to send their NAKs.

The protocols presented above do not have any feedback suppression mechanisms. In case of multicast communication, we may have several NAKs coming from different receivers to the sender for the same block causing a feedback implosion at the sender. A feedback suppression mechanism, like the one proposed in [FJL+97], is useful to reduce the processing load of the received feedbacks in the sender. However, it adds an extra delay in case of retransmission due to its mechanism for feedback suppression.

Variable	**Definition**
D	Packet delay
D_d	Transmission time of a data packet
D_c	Transmission time of a control packet
X_c	Control packet (NAK, POLL) processing time at sender
X_e	Encoding delay per packet at sender
T_s	The timer value at sender
Y_d	Data packet processing time at receiver
Y_c	Control packet (NAK, POLL) processing time at receiver
Y_e	Decoding delay per packet at receiver
N_r	Number of transmission rounds for a packet
A	Number of received NAKs at the sender
R	Number of receivers
l	Number of lost packets
L_B	Packet size in Bytes

Table 6.1: The definition of the variables used in the delay analysis

The list of all variables involved in the delay analysis of a packet is presented in Table 6.1. Note that in both protocols, the packets are buffered at the receiver side. The receiver

delivers the packets to the higher layer if it receives all the k packets correctly or if it receives a POLL message. In other words, a packet can not be delivered to the higher layer before the end of the reception of its block. Therefore, the delay of each packet is nearly equivalent to the delay of its block since it must wait for the reception of all the packets in its block before going to the higher layer. Moreover in case of packet losses, the receiver has to wait for the transmission of the next block. Therefore, we calculate the average delay of a block in the following. We also ignore the effect of propagation delay in our analysis.

Let us begin with protocol P1 which uses only an ARQ mechanism. Considering that each packet is transmitted $E[M]$ times in average, the total delay due to the transmission of a block is $kE[M]D_d$. Since the protocol does not work in a continuous way, the sender has to either receive R NAKs or make a timeout in order to continue the transmission of the next block. Therefore, a packet can not be retransmitted immediately. We define *Round Trip Delay* (RTD) as the minimum time that it takes between the end of the transmission of a block and the beginning of the transmission of the next block. Note that this minimum time corresponds to the situation where the sender receives R NAKs before the expiration of its timer. In case the sender receives less than R NAKs, it has to wait for the timeout. The timer value must be high enough to allow all receivers to send their NAKs. The sender starts the timer right after the transmission of the POLL message. Therefore, the retransmission only takes place after the POLL processing delay at the sender, the POLL transmission time and the RTD in the best case.

In order to calculate the RTD, we have to take into account the processing time of the POLL message at the receiver, $E[Y_c]$, the processing time of a NAK message at the receiver, $E[Y_c]$, the NAK transmission time, D_c, and finally the NAK processing time at the sender, $RE[X_c]$. Therefore, we have:

$$RTD = RE[X_c] + 2E[Y_c] + D_c$$

Recalling that every receiver sends only one NAK for k packets, the probability to have a NAKs, $\Phi(A = a)$, is the probability to have only a receivers that have lost at least one packet among the k transmitted packets. This probability can be calculated as follows:

$$\Phi(A = a) = \binom{R}{a}(1 - (1 - p)^k)^a (1 - p)^{(R-a)k} \tag{6.10}$$

The average delay of a packet in protocol P1 is then determined as follows. Note that the RTD and T_s are the minimum and the maximum times that it takes before the sender starts transmitting the next block. The $E[X_c] + D_c$ in the following formula corresponds to the POLL processing and transmission delays.

$$\begin{aligned} E[D] =& kE[M]D_d + kE[Y_d] \\ &+ (E[M] - 1)\Big(E[X_c] + D_c + \Phi(A = R)RTD \\ &+ (1 - \Phi(A = R) - \Phi(A = 0))T_s\Big) \end{aligned} \tag{6.11}$$

The delay of protocol P2 is essentially the same but with the addition of coding delays. Recall that if a packet is correctly received no decoding is necessary since we use systematic codes. Decoding is only used when a packet is lost but there are enough redundant packets in order to recover. We take P_d as the probability of decoding which is the probability to have at least one receiver which has lost a packet but it has enough redundant packets to recover from the loss (at least k packets out of the other $n-1$ packets are correctly received). This probability can be calculated as:

$$\begin{aligned} P_d &= \sum_{r=1}^{R} \binom{R}{r} \left(p \sum_{j=0}^{h-1} \binom{n-1}{j} p^j (1-p)^{n-j-1} \right)^r (1-p)^{R-r} \\ &= (1-q)^R - (1-p)^R \end{aligned} \tag{6.12}$$

We define $E[N_r]$ as the average number of transmission rounds necessary for a packet to be correctly received by all receivers. In fact, $E[N_r]$ is the average number of transmissions required for a packet to be correctly received by all receivers without taking into account the coding overhead:

$$E[M] = \frac{n}{k} E[N_r] \tag{6.13}$$

Rizzo has calculated the time required to produce h parity packets in [Riz97] as follows:

$$X_e = \frac{kL_B}{c_e}(n-k) \tag{6.14}$$

where L_B is the packet size in Bytes and c_e is the encoding constant in Byte/sec. In the same way, the decoding time is $Y_e = \frac{kL_B}{c_d} l$ where l is the number of lost packets and c_d is the decoding constant in Byte/sec. Assuming that the number of lost packets in a coded block is equal to the number of redundant packets in a block, the decoding delay that each packet undergoes is:

$$Y_e = \frac{kL_B}{c_d}(n-k) \tag{6.15}$$

The last thing that we have to calculate is the probability density function of the generated NAKs which can be easily found replacing p by q in equation (6.10):

$$\Phi(A=a) = \binom{R}{a} (1-(1-q)^k)^a (1-q)^{(R-a)k} \tag{6.16}$$

The overall average delay of protocol P2 can be determined as below. The RTD is exactly the same as in protocol P1. We have simplified the decoding delay by assuming that the

number of lost packets in a block is equivalent to the number of redundant packets of the block. The implications of this simplification are that we calculate the decoding delay in the worst case and that only k packets among the n transmitted ones are correctly received at the last transmission round.

$$\begin{aligned} E[D] =& nE[N_r]D_d + E[N_r]X_e + kE[Y_d] + P_dY_e \\ &+ (E[N_r]-1)\Big(E[X_c] + D_c + \Phi(A=R)RTD \\ &+ (1-\Phi(A=R)-\Phi(A=0))T_s\Big) \end{aligned} \tag{6.17}$$

We took the same RSE codes that we took for the evaluation of efficiency. Data packet size is 54 bytes and control packet size is 9 bytes. The data rate is fixed at 20 Mb/s. With these values, we have $D_c = 3.6\mu sec$ and $D_d = 21.6\mu sec$. From [Riz97], we find $c_d \approx c_e = 4.6884$ MByte/sec. The timer value is fixed at $T_s = 1.2RTD$. Figure 6.4 depicts the average delay as a function of different RTD values. In this figure, a block size of 50 packets is used for P1 protocol. For the other figures, we fix the control packet processing time at the sender and the receiver at $E[X_c] = E[Y_c] = 50\mu sec$ and the data packet processing time at the receiver at $E[Y_d] = 250\mu sec$.

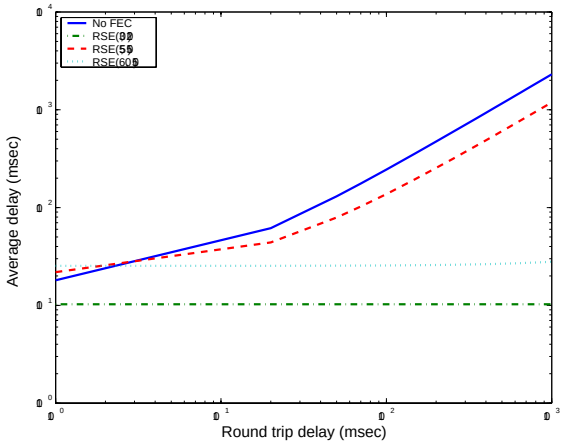


Figure 6.4: Average delay as a function of RTD, $R = 1000$, $e = 10^{-4}$

Figure 6.4 is quite expressive. FEC reduces the average delay of each packet in most cases and therefore it is useful for real-time applications. However, RTD or more generally, the time that a retransmission takes is an important factor which affects the suitability of FEC as a mechanism to decrease the delay. FEC is more interesting in cases where the retransmission takes some time. In a wireless system, the MAC protocol used to access the radio channel affects the RTD. For example in a TDMA system, a packet can only be retransmitted to an MT in its dedicated slots while in an ALOHA system, a packet can be retransmitted right away. In our delay calculation, we ignored the extra delay due to the MAC protocol logic. It is clear that this delay increases the RTD.

Figure 6.5 depicts the average delay as a function of the bit error rate. In the first plot, we only vary the number of redundant packets and we keep the number of data packets fixed

($k = 50$). The ARQ scheme has the same block size as the other schemes (50 data packets) in the first plot. For low bit error rates, the codes with more redundant packets have higher average delays. This effect is mostly due to their encoding and decoding delays since these delays are proportional to the number of redundant packets. The size of the coded block also plays an important role in the average delay. Long blocks cause a higher delay compared to short blocks. This effect is more evident in the second plot where $RSE(20, 10)$ provides the lowest delay. We observe that the increase of the number of redundant packets causes a more stable delay for high bit error rates. However for very high bit error rates, all the schemes require a large number of retransmissions. In this case, the effect of retransmission delay becomes the dominant factor and all the curves overlap.

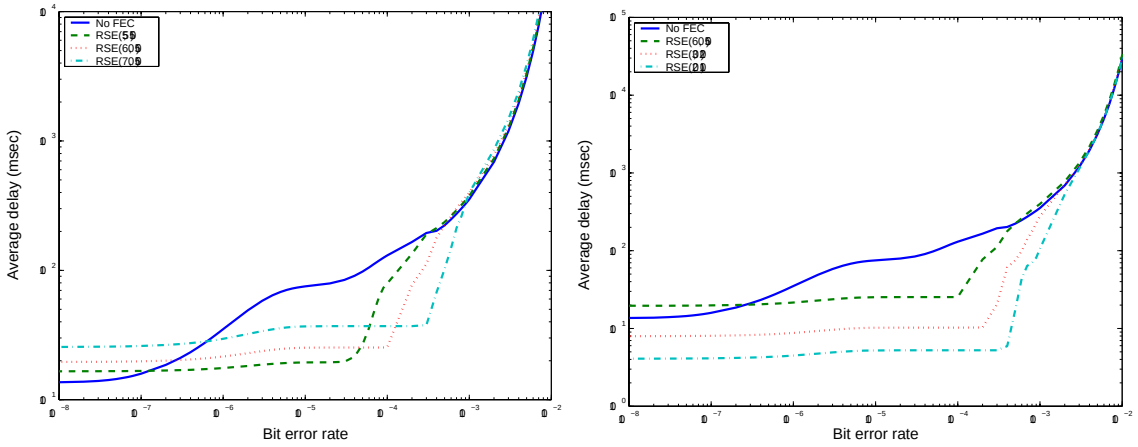


Figure 6.5: Average delay as a function of bit error rate, R=1000

The second plot maintains a constant number of redundant packets but changes the data block size k. The ARQ scheme has again a data block size of 50 packets in this plot. As it can be seen, the codes with a smaller block size has lower delays for low bit error rates. This is due to the fact that the transmission time, propagation time and the processing time of a short block takes less time compared to long blocks. In the second plot, the total encoding and decoding delays are the same for all the schemes since they all have the same number of redundant packets. The only dominant factor is the different transmission and processing delays due to their different block sizes.

Figure 6.6 depicts the average delay as a function of the number of wireless receivers. The bit error rate is fixed at 10^{-4}. Once again, we observe that the codes with higher number of redundant packets provide a more stable average delay. However, for a high number of receivers, the number of feedback messages coming to the sender increases. This increase will cause an increase in the processing time of the sender which in turn may affect seriously the average delay. As it can be seen in the figure, $RSE(60, 50)$ performs better than $RSE(30, 20)$ since the sender receives a lower number of NAKs (one NAK for a block of 50 packets compared to one NAK per 20 packets in case of $RSE(30, 20)$).

In general, the numerical analysis show that the extra delay of encoding and decoding is tolerable for most applications. The use of FEC reduces the average delay that each packet undergoes. It also causes the average delay to stay constant for at least a certain range of bit error rates. This observation makes us conclude that the use of FEC can also help reduce the

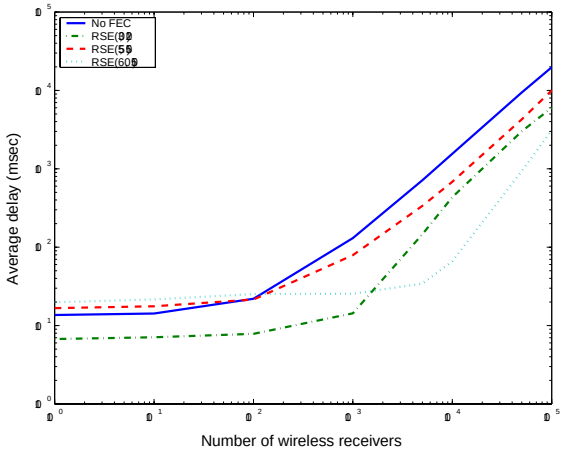


Figure 6.6: Average delay as a function of number of wireless receivers, e=10^{-4}

variations of delay (jitter) of each packet which is an important QoS parameter for real time applications.

6.4 Performance Evaluation of FEC in a GE Model

6.4.1 Gilbert-Elliot model

Two state Markov models have been extensively used in the literature to capture the bursty nature of the error sequences generated by a wireless channel. Previous studies [ZRM95], [Wan94] show that a first order Markov chain such as a two state Markov model provides a good approximation of the error process in fading channels. The two state Markov model is widely known as Gilbert-Elliot (GE) model. The model was first used by Gilbert [Gil60]. Elliot generalized slightly the Gilbert model in [Ell63]. We take a GE model, as shown in Figure 6.7, to characterize the error sequences in a fading channel.

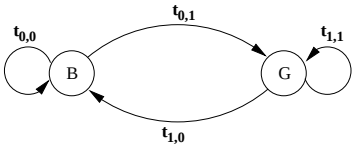


Figure 6.7: Gilbert-Elliot model

The model consists of two states. State G corresponds to the Good state where errors occur with a low probability e_G. State B corresponds to the Bad state where errors occur with a high probability e_B. One of the advantages of this model is the facility to map its parameters to real physical quantities in case of a Rayleigh fading channel. To obtain such a relation, we note that Rayleigh fading results in an exponentially distributed distortion of the signal. The probability density function of the *Signal to Noise Ratio* (SNR), λ, can be written as follows [Pro89]:

$$f(\lambda) = \frac{1}{\bar{\lambda}} exp(-\frac{\lambda}{\bar{\lambda}}) \qquad \lambda > 0 \tag{6.18}$$

where $\bar{\lambda}$ is the average SNR.

If the received SNR is above a certain threshold, λ_T, the channel is in the Good state. It is in the Bad state if the received SNR is below λ_T. Using the *level crossing rate* and the SNR density function, the parameters of the model can be found in terms of physical quantities [SDH96] [WM95]. Assuming that the channel fades slowly with respect to the symbol interval, T, the transition probabilities of the Markov chain can be calculated as:

$$t_{0,1} = \frac{f_d T \sqrt{2\pi \frac{\lambda_T}{\bar{\lambda}}}}{exp(\frac{\lambda_T}{\bar{\lambda}}) - 1} \tag{6.19}$$

$$t_{1,0} = f_d T \sqrt{2\pi \frac{\lambda_T}{\bar{\lambda}}} \tag{6.20}$$

$$t_{0,0} = 1 - t_{0,1} \tag{6.21}$$

$$t_{1,1} = 1 - t_{1,0} \tag{6.22}$$

where f_d is the maximum doppler frequency given by $f_d = \frac{v f_c}{c}$ with v the vehicle speed, f_c the carrier frequency and c the speed of light ($3 \times 10^8 m/s$). The steady state probabilities π_G and π_B can be found as:

$$\pi_G = \int_{\lambda_T}^{\infty} f(\lambda) d\lambda = exp(\frac{-\lambda_T}{\bar{\lambda}}) \tag{6.23}$$

$$\pi_B = \int_{0}^{\lambda_T} f(\lambda) d\lambda = 1 - exp(\frac{-\lambda_T}{\bar{\lambda}}) \tag{6.24}$$

The error probabilities e_G and e_B of each state can be related to the received SNR according to the modulation scheme used in the system. They are given by

$$e_G = \frac{\int_{\lambda_T}^{\infty} f(\lambda) e_m(\lambda) d\lambda}{\int_{\lambda_T}^{\infty} f(\lambda) d\lambda} = \frac{1}{\pi_G} \int_{\lambda_T}^{\infty} f(\lambda) e_m(\lambda) d\lambda \tag{6.25}$$

$$e_B = \frac{\int_{0}^{\lambda_T} f(\lambda) e_m(\lambda) d\lambda}{\int_{0}^{\lambda_T} f(\lambda) d\lambda} = \frac{1}{\pi_B} \int_{0}^{\lambda_T} f(\lambda) e_m(\lambda) d\lambda \tag{6.26}$$

$e_m(\lambda)$ in the above formulas represents the error probability as a function of the received SNR. $e_m(\lambda)$ depends on the modulation scheme used. For a *Binary Phase Shift Keying* (BPSK) scheme, we have:

$$e_m(\lambda) = 1 - F(\sqrt{2\lambda})$$

where

$$F(\lambda) = \int_{-\infty}^{\lambda} \frac{1}{\sqrt{2\pi}} exp(-\frac{x^2}{2}) dx$$

Simplified expressions for e_G and e_B are provided in [WM95] for a BPSK scheme. The average error rate of the model can be found as $e = e_G\pi_G + e_B\pi_B$.

It is important to note that the correlation property of the fading process depends only on f_dT. If the value $f_dT < 0.1$, the fading process is very correlated and is considered as slow fading. In this case, the assumption that the losses are independent is not correct. For the values of $f_dT > 0.2$, two samples of the channel are almost independent and the fading process is considered as fast fading [ZRM95].

6.4.2 The Effect of FEC on Efficiency

In order to investigate the effect of packet level FEC, we are interested to model the process of successful or unsuccessful packet transmission. [ZRM98] showed that a Markov approximation for a packet loss process is a good model for a broad range of parameters. In fact for typical data rates (e.g. more than 64 Kb/s) and for environments commonly considered (e.g. carrier frequency of about 1-2 GHZ and typical pedestrian and vehicular speeds), we can assume that the channel is constant during a packet interval. Therefore, without loss of generality, we can apply the same GE model to packets with T taken as packet interval and p_G and p_B as packet loss rates in Good and Bad states respectively. Since the channel remains at the same state during a packet interval, the packet loss rates of each state can be calculated as in a BSC model.

$$p_G = 1 - (1 - e_G)^L \tag{6.27}$$

$$p_B = 1 - (1 - e_B)^L \tag{6.28}$$

Let us first consider the scenario where a sender multicasts data to R receivers using an ARQ scheme. The sender retransmits the original packet if there is at least one receiver that has not received the packet correctly. Generally the sender is not aware of a packet loss unless it receives a negative feedback from one of the receivers. In this case, it can only retransmit the lost packet after a certain time t due to the retransmission delay as explained in the previous section. In our analysis, we assume that the channel remains at the same state during the time interval $T + t$. It is clear that if $T + t$ is longer than the correlation time of the channel, the assumption that the channel stays at the same state during the $T + t$ is not correct.

We define L_r as the number of times that a packet gets lost by a receiver. We assume that the state transitions occur at the beginning of a time slot of unit length and then a packet is transmitted. The probability that a packet gets lost in a receiver exactly l times $P(L_r = l)$ is the sum of $P_G(L_r = l)$, the probability that a packet gets lost in a receiver exactly l times with the channel ending in state G, and $P_B(L_r = l)$, the probability that a packet gets lost in a receiver exactly l times with the channel ending in state B.

$$P(L_r = l) = P_G(L_r = l) + P_B(L_r = l), \tag{6.29}$$

$$P_G(L_r = l) = \begin{cases} (1 - p_G)\pi_G & l = 0, \\ p_G\pi_G & l = 1, \\ P_G(L_r = l-1)t_{1,1}p_G + P_B(L_r = l-1)t_{0,1}p_G & l = 2, 3, ... \end{cases}$$

$$P_B(L_r = l) = \begin{cases} (1 - p_B)\pi_B & l = 0, \\ p_B\pi_B & l = 1, \\ P_G(L_r = l-1)t_{1,0}p_B + P_B(L_r = l-1)t_{0,0}p_B & l = 2, 3, ... \end{cases}$$

Again, M_r is the number of transmissions required for a correct reception of a packet by a receiver r and M is the number of transmissions required for a correct reception of a packet by all receivers. The average number of transmissions, $E[M]$, as well as the efficiency of the scheme, Eff, can be calculated as follows:

$$P(M_r \leq m) = 1 - P(L_r = m), \tag{6.30}$$

$$P(M \leq m) = (1 - P(L_r = m))^R \tag{6.31}$$

$$E[M] = \sum_{m=1}^{\infty} mP(M = m) = \sum_{m=1}^{\infty} P(M \geq m) \tag{6.32}$$

$$Eff = \frac{1}{E[M]} = \frac{1}{\sum_{m=1}^{\infty} [1 - (1 - P(L_r = m-1))^R]} \tag{6.33}$$

Now, we consider the ARQ/FEC case where the sender uses an RSE code which generates a coded block of n packets containing k data packets and h redundant packets. In this case, the sender sends k original packets followed by h redundant ones. Each receiver can recover from losses if it receives correctly k packets out of the $k + h$ transmitted packets.

[YW95] calculated the probability to have i errors in j transmissions in a Gilbert-Elliot model using recursion. Using the same approach, we can calculate the probability to have i packet losses among j transmitted packets, $P(i, j)$, in a Gilbert-Elliot model. Let $P_B(i, j)$ be the probability to have i packet losses among j transmitted packets with the channel ending in state B and $P_G(i, j)$ be the probability to have i packet losses among j transmitted packets with the channel ending in state G. As before, we assume that state transitions occur at the beginning of a time slot of unit length and then a packet is transmitted. We have:

$$
\begin{aligned}
P(i,j) =& P_G(i,j) + P_B(i,j) \\
P_G(i,j) =& P_G(i,j-1)t_{1,1}(1-p_G) \\
&+ P_B(i,j-1)t_{0,1}(1-p_G) \\
&+ P_G(i-1,j-1)t_{1,1}p_G \\
&+ P_B(i-1,j-1)t_{0,1}p_G \\
P_B(i,j) =& P_B(i,j-1)t_{0,0}(1-p_B) \\
&+ P_G(i,j-1)t_{1,0}(1-p_B) \\
&+ P_B(i-1,j-1)t_{0,0}p_B \\
&+ P_G(i-1,j-1)t_{1,0}p_B
\end{aligned}
\tag{6.34}
$$

for $j = 1, 2, 3...$ and $i = 0, 1, 2, ..., j$.

Let's define $Q(L_r = l)$ as the probability that a packet gets lost by a receiver exactly l times when using FEC. This probability is again the sum of $Q_G(L_r = l)$, the probability that a packet gets lost by a receiver exactly l times with the channel ending in state G, and $Q_B(L_r = l)$, the probability that a packet gets lost by a receiver exactly l times with the channel ending in state B. In the presence of FEC, a packet is retransmitted if it is lost by the FEC receiver and if more than $h-1$ out of the other $n-1$ packets of the coded block are lost. In the same way, a packet is considered to be correctly received if it has not been lost or if it has been lost but there are at least $h-1$ packets out of the other $n-1$ packets of the coded block that have been correctly received. Once again, we assume that the channel does not change its state during the interval $T+t$ where t corresponds to the time between the end of the transmission of the last packet of a coded block and the beginning of the transmission of the first packet of the next coded block.

$$
Q(L_r = l) = Q_G(L_r = l) + Q_B(L_r = l), \tag{6.35}
$$

$$
Q_G(L_r = l) = \begin{cases}
\sum_{i=0}^{h-1} \Big[P_G(i,n-1)t_{1,1}p_G + P_B(i,n-1)t_{0,1}p_G \Big] + \\
\sum_{i=0}^{n-1} \Big[P_G(i,n-1)t_{1,1}(1-p_G) + P_B(i,n-1)t_{0,1}(1-p_G) \Big] & l = 0 \\
\sum_{i=h}^{n-1} \Big[P_G(i,n-1)t_{1,1}p_G + P_B(i,n-1)t_{0,1}p_G \Big] & l = 1, 2, ...
\end{cases}
$$

$$
Q_B(L_r = l) = \begin{cases}
\sum_{i=0}^{h-1} \Big[P_G(i,n-1)t_{1,0}p_B + P_B(i,n-1)t_{0,0}p_B \Big] + \\
\sum_{i=0}^{n-1} \Big[P_G(i,n-1)t_{1,0}(1-p_B) + P_B(i,n-1)t_{0,0}(1-p_B) \Big] & l = 0 \\
\sum_{i=h}^{n-1} \Big[P_G(i,n-1)t_{1,0}p_B + P_B(i,n-1)t_{0,0}p_B \Big] & l = 1, 2, ...
\end{cases}
$$

The initial values for $P(i,j)$ in the above formulas are:

$$P_G(0,0) = \begin{cases} \pi_G & l = 0,1 \\ Q_G(L_r = l-1) & l = 2,3,... \end{cases}$$
$$P_B(0,0) = \begin{cases} \pi_B & l = 0,1 \\ Q_B(L_r = l-1) & l = 2,3,... \end{cases}$$

and $P_B(i,0) = P_G(i,0) = 0$ for $i \neq 0$. It is clear that with these initial values, all numerical values are steady state results. The efficiency is then calculated by using $Q(L_r = l)$ as in equation (6.35) and by taking into account the coding overhead:

$$Eff = \frac{1}{E[M]} = \frac{k}{n} \frac{1}{\sum_{m=1}^{\infty} [1 - (1 - Q(L_r = m-1))^R]} \tag{6.36}$$

For our analysis, we take a pedestrian speed of 1 m/s corresponding to a doppler frequency of 17.3 Hz for a carrier frequency of 5.2 GHz . We take a data rate of 20 Mb/s and a packet size of 54 bytes corresponding to a packet interval of $T = 21.6\mu sec$. For these values, we have a slow fading channel ($f_d T = 0.0003744$). One of the parameters of the model which may affect the overall performance analysis is the threshold SNR, λ_T. The choice of λ_T affects the error probabilities, e_G and e_B, as well as the steady state probabilities of each state, π_G and π_B. Figure 6.8 depicts the effect of threshold SNR on the efficiency of the ARQ scenario. We used a BPSK modulation scheme for this figure.

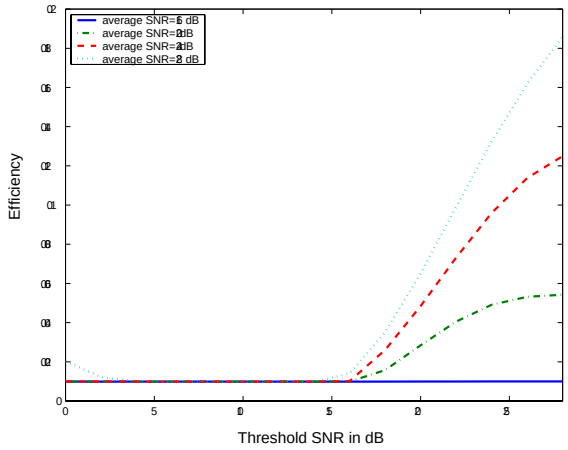


Figure 6.8: Efficiency as a function of threshold SNR, R=1000

We observe that for an average SNR of 16 dB, the efficiency remains constant while for other average SNRs the efficiency starts increasing after a λ_T of around 15dB. One direct impact of increasing the threshold SNR is the decrease of e_G which in turn causes an increase in efficiency. In fact by increasing λ_T, e_G starts decreasing until it reaches zero. Further analysis shows that a threshold SNR of about 15 dB corresponds to an $e_G \approx 0$ for an average

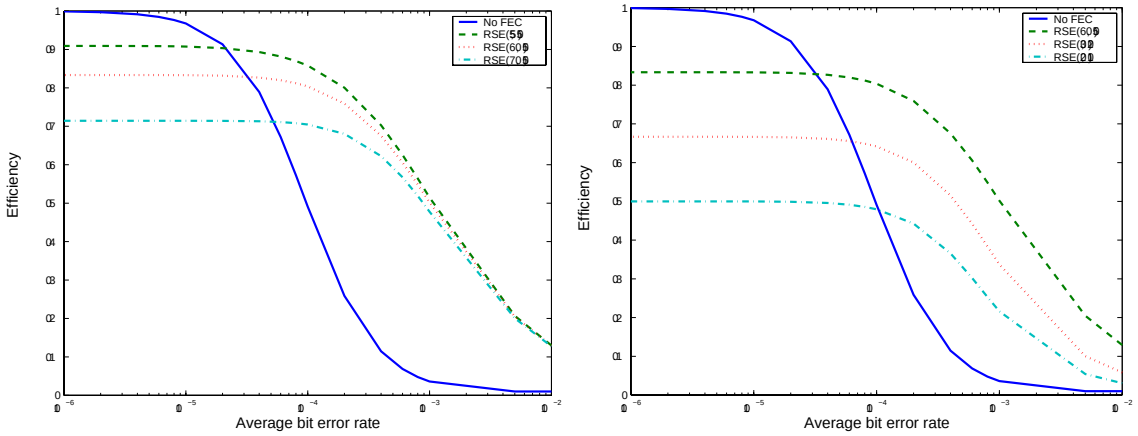


Figure 6.9: Efficiency as a function of average bit error rate, $f_d T = 0.0003744$, $R = 1000$

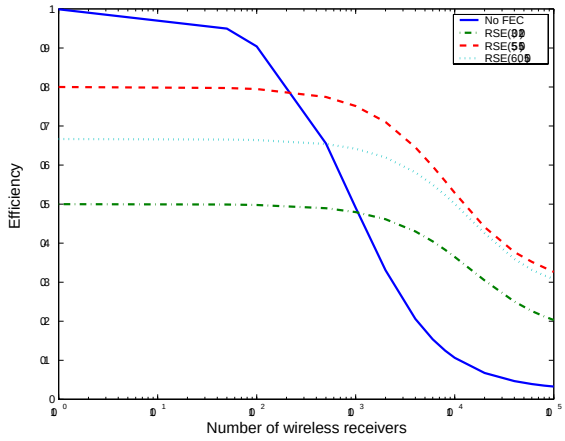
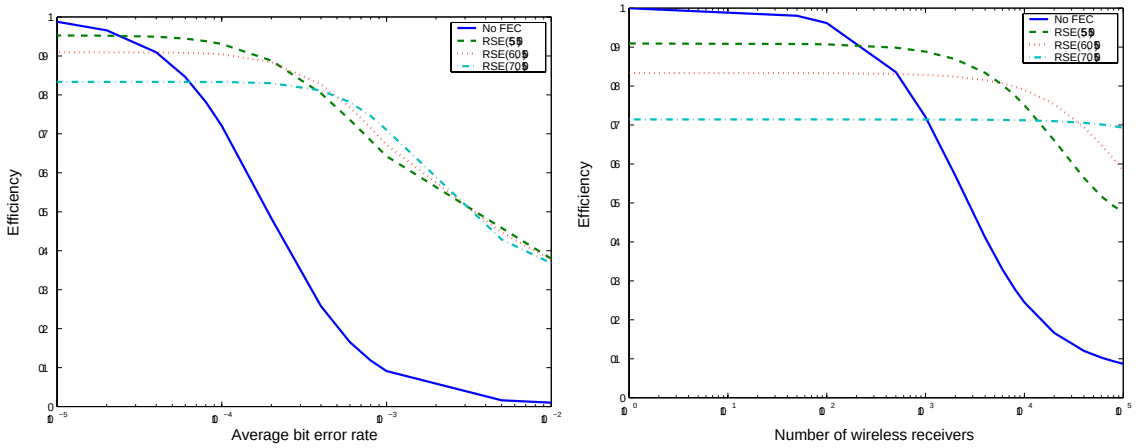


Figure 6.10: Efficiency as a function of number of wireless receivers, $f_d T = 0.0003744$, $e = 10^{-4}$

SNR of 20, 24 and 28 dB while for an average SNR of 16dB we need to increase the threshold SNR beyond the range of threshold SNRs depicted in the figure in order to obtain a zero error probability at the Good state. This explains why this curve remains stable in Figure 6.8. For other curves, the increase of threshold SNR beyond the limit of 15dB has no effect on e_G but it decreases e_B making the overall efficiency better. We also observe that for higher average SNRs and for the same threshold SNR, we get better results. This is because, higher average SNRs causes an increase in π_G which in turn decreases the average bit error rate of the system and improves the efficiency.

We take a model with $e_G \approx 0$ and $e_B \approx 1$. In this case the average error rate of the model will be $e = 1 - exp(\frac{-\lambda_T}{\lambda})$. Knowing e gives us easily the ratio $\frac{\lambda_T}{\lambda}$. Note that the transition probabilities of the model as well as its steady state probabilities depend only on this ratio. The advantage of such a choice is that the parameters of the model become independent of the modulation scheme and packet length.

Figure 6.9 shows the efficiency as a function of the bit error rate in a group of 1000



(a) $R = 1000$ (b) $e = 10^{-4}$

Figure 6.11: Efficiency as a function of average bit error rate and number of wireless receivers, $f_dT = 0.001$

wireless receivers. Figure 6.10 depicts the efficiency as a function of the number of wireless receivers with an average bit error rate of 10^{-4}. Contrary to what we were thinking, these figures show us that the increase of number of redundant packets or the decrease of data block size can not increase the efficiency anymore and that the code with the largest data block size and the lowest redundancy has the best efficiency. However, we think that this is due to the fact that the parameters we have chosen cause a long fading interval. In order to observe the effect of coding, we have to either increase the coded block size in order to cover a time interval larger than the fading interval or to change the parameters f_dT in order to have a shorter fading period.

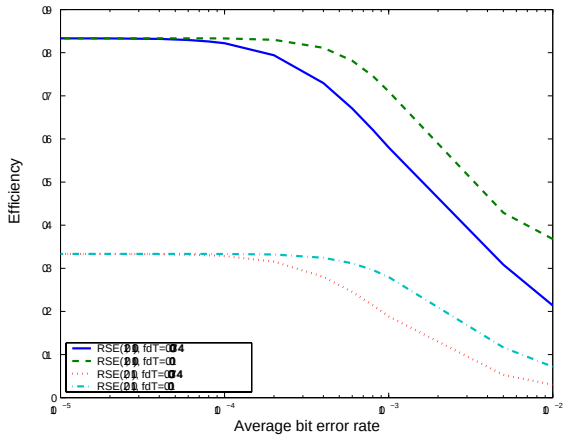


Figure 6.12: Efficiency as a function of average bit error rate for different values of f_dT, R=1000

Figure 6.11 proves our conclusion, showing that the increase of redundant packets does improve the efficiency but with the condition that the coded block covers a longer time

interval than the fading period. However, the decrease of data block size must be made carefully. Decreasing k improves the efficiency as long as the new data block size is still large enough to span the fading period. If this is not the case, the decrease of data block size must be avoided. In order to see the effect of f_dT on the overall performance of coding, Figure 6.12 depicts the efficiency of two different codes for different values of f_dT. From this figure, we conclude that FEC performs better when errors are less correlated. In case of high correlation, codes with larger coded blocks can help increase efficiency.

6.4.3 The Effect of FEC on Packet Loss Rate

We have already calculated the probability of a receiver to lose a packet l times in equation (6.29) in case of an ARQ scheme. The probability of a receiver to receive a packet correctly after its first transmission is then $P(L_r = 0)$. Knowing $P(L_r = 0)$, the packet loss rate, PLR, in case of ARQ can be determined as:

$$PLR = 1 - P(L_r = 0)^R \tag{6.37}$$

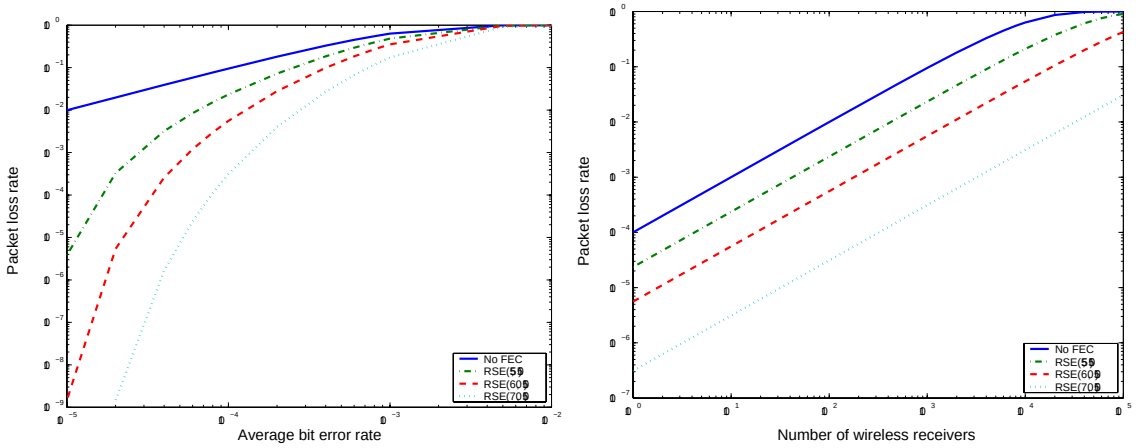
Equation (6.35) calculates the probability that a receiver looses a packet l times, $Q(L_r = l)$, in the case of FEC. From there, the PLR for an ARQ/FEC scenario can be found as:

$$PLR = 1 - Q(L_r = 0)^R \tag{6.38}$$

Figure 6.13 depicts the effect of average bit error rate and the number of wireless receivers on the packet loss rate of different scenarios. We took an $f_dT = 0.001$. We observe that the increase of redundant packets can help decreasing the packet loss rate in the presence of high average bit error rates and number of receivers. We do not investigate the effect of shorter data block size since it has been shown to be inefficient with long fading intervals.

6.4.4 The Effect of FEC on Delay

Once again, we consider protocols P1 and P2. In order to calculate the average delay of P1, we need to know the average number of transmissions for a packet $E[M]$ taking into account that P1 sends k packets at a time. This consideration is required since the timing information is important in a GE model. The channel state in instant t may be different from the channel state in instant $t + t_0$. We have already calculated $E[M]$ in equation (6.33) but with the assumption that the sender transmits one packet at a time. In this case, $P(L_r = l)$ is as in equation (6.29). If the sender sends k packets at a time as in protocol P1, $P(L_r = l)$ is found as follows:



(a) $R = 1000$ (b) $e = 10^{-4}$

Figure 6.13: Packet loss rate as a function of average bit error rate and number of wireless receivers, $f_d T = 0.001$

$$P(L_r = l) = P_G(L_r = l) + P_B(L_r = l), \tag{6.39}$$

$$P_G(L_r = l) = \begin{cases} \sum_{i=0}^{k-1} \Big[P_G(i, k-1) t_{1,1}(1 - p_G) + P_B(i, k-1) t_{0,1}(1 - p_G) \Big] & l = 0 \\ \sum_{i=0}^{k-1} \Big[P_G(i, k-1) t_{1,1} p_G + P_B(i, k-1) t_{0,1} p_G \Big] & l = 1, 2, ... \end{cases}$$

$$P_B(L_r = l) = \begin{cases} \sum_{i=0}^{k-1} \Big[P_G(i, k-1) t_{1,0}(1 - p_B) + P_B(i, k-1) t_{0,0}(1 - p_B) \Big] & l = 0 \\ \sum_{i=0}^{k-1} \Big[P_G(i, k-1) t_{1,0} p_B + P_B(i, k-1) t_{0,0} p_B \Big] & l = 1, 2, ... \end{cases}$$

where the initial values for $P(i, j)$ are:

$$P_G(0, 0) = \begin{cases} \pi_G & l = 0, 1 \\ P_G(L_r = l - 1) & l = 2, 3, ... \end{cases}$$

$$P_B(0, 0) = \begin{cases} \pi_B & l = 0, 1 \\ P_B(L_r = l - 1) & l = 2, 3, ... \end{cases}$$

and $P_B(i, 0) = P_G(i, 0) = 0$ for $i \neq 0$. With this $P(L_r = l)$, we can find $E[M]$ in case of protocol P1. The only remaining probability that we have to calculate for the GE model is the probability density function of the number of generated NAKs which can be found as follows:

$$\Phi(A=a) = \binom{R}{a}(1-P(0,k))^{a}(P(0,k))^{(R-a)} \tag{6.40}$$

$P(0,k)$ is the probability to have zero losses in k packet transmissions and $(1-P(0,k))$ is the probability to have at least one loss in k packet transmissions. The delay of the protocol P1 can be calculated as in equation (6.11) using $P(L_r=l)$ from equation (6.39) to calculate $E[M]$ and $\Phi(A=a)$ from equation (6.40).

In case of protocol P2, the probability of decoding, P_d, can be found using $Q(L_r=l)$ as in equation (6.35) and $P(L_r=l)$ as in equation (6.39):

$$P_d = Q(L_r=0)^R - P(L_r=0)^R \tag{6.41}$$

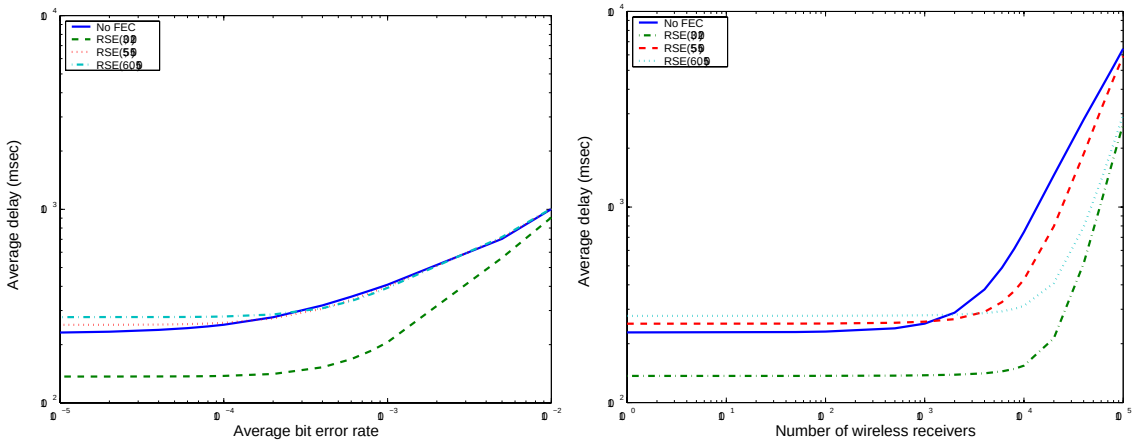
We also have to calculate the probability density function of the generated NAKs in case of P2. It is important to note that in case of FEC, a packet is considered lost if there are more than h losses in n transmitted packets. Therefore, the probability to have zero NAKs at the end of the transmission of a coded block is the probability to have not more than h losses. $\sum_{i=0}^{h} P(i,n)$ represents this probability. $P(i,n)$ represents the probability to have i packet losses in a coded block of n packets. It can be computed using equation (6.34):

$$\Phi(A=a) = \binom{R}{a}\left(1-\sum_{i=0}^{h} P(i,n)\right)^{a}\left(\sum_{i=0}^{h} P(i,n)\right)^{(R-a)} \tag{6.42}$$

The delay of protocol P2 can be found as in equation (6.17) using equations (6.41) and (6.42).

We take an $f_dT = 0.001$ for our delay analysis with a data rate of 100 Kb/sec. Data packet size and control packet sizes are again 54 and 9 bytes. The control packet processing time at the sender and the receiver are fixed at $E[X_c] = E[Y_c] = 50\mu sec$. The data packet processing time at the receiver is $E[Y_d] = 250\mu sec$. The timer is again $T_s = 1.2RTD$. The coding constants are $c_d \approx c_e = 4.6884$ MByte/s. Once again the propagation delay is ignored. One simplification that we have made in the depicted results is that we have used steady state probabilities as the initial values of the $P(0,k)$ and $P(i,n)$ in equations (6.40) and (6.42).

The first plot in Figure 6.14 depicts the average delay as a function of bit error rate. The ARQ scheme has a block size of 50 packets in both plots. The second plot shows the average delay as a function of wireless receivers for a bit error rate of 10^{-4}. We can observe that the use of FEC can help reduce the average delay of each packet. For high bit error rates and high number of receivers, it is better to use more redundant packets. The numerical results obtained for efficiency and packet loss rate showed that the decrease of the number of data packets must be avoided unless the new coded block size can span the whole fading interval. However for the average delay, we observe that the $RSE(55,50)$ and $RSE(60,50)$ only improves the average delay slightly while $RSE(30,20)$ provides the lowest delay. This is once again due to the small block size of this code. As stated before, a small block takes less time to transmit and process than a long block.



(a) $R = 1000$ (b) $e = 10^{-4}$

Figure 6.14: Average delay as a function of average bit error rate and number of wireless receivers, $f_dT = 0.001$

6.5 Conclusion

In this chapter, we studied the effect of coding on three QoS metrics: bandwidth, average delay and the packet loss rate. We took two different error models, BSC and GE. For each model, we calculated these metrics using different error control protocols. The first protocol corresponds to a pure ARQ scheme while other protocols correspond to hybrid ARQ/FEC schemes. All the hybrid ARQ/FEC protocols use Reed-Solomon erasure codes but with different coding parameters. For all scenarios, we considered a multicast communication mode where the traffic is transmitted toward a set of receivers. This will allow us to generalize our analysis to a different number of receivers. We presented several numerical results in this chapter. Based on these results, we can take the following conclusions:

- Hybrid ARQ/FEC outperforms ARQ in most cases.
- There is no unique best code for a hybrid ARQ/FEC protocol.
- In case of a BSC model, choosing a code is a function of the bit error rate of the channel, packet length, number of receivers and the desired QoS. If we assume that C represents the best code, we have:

  $C = \mathcal{F}(e, L, R, QoS)$

- In case of a GE model, we have to consider the effect of threshold SNR and f_dT as well. Therefore, we have:

  $C = \mathcal{F}(e, \lambda_T, f_dT, L, R, QoS)$

  Note that the average bit error rate of a GE model depends on the average SNR and the modulation scheme used.

- For very high bit error rates, even a coding scheme can not help.
- Normally for low bit error rates and a low number of receivers, a coding scheme with a high number of data packets and a small number of redundant packets is a good starting point. As the bit error rate and the number of receivers increase, we have to increase the number of redundant packets in order to have a reasonable performance. In case the maximum number of available redundant packets is not sufficient to guarantee an acceptable level of performance, the number of data packets must be decreased while the number of redundant packets must be maintained at its maximum. While this solution always leads to better results in case of BSC model, it may not improve the performance in case of GE model due to the correlated nature of errors in this model. In a GE model, coding can improve the performance if the coded block size is larger than the correlation time of the channel. Supposing that the channel stays at the Bad state during the whole interval of a code, all packets will be lost anyway. If the channel is in Good state, all packets will be correctly received with or without coding. With such a condition, a simple retransmission scheme will be more efficient in both states.

Although the obtained results showed the advantages of FEC in most cases, we observed a high performance degradation in case of GE model. We think that this is mostly due to our two-state model which does not provide the bit error rate variations in a real wireless channel. We generalize this model in Chapter 7 in order to provide a more realistic model for wireless channels.

Summarizing all, when choosing a code, the desired QoS criteria of the session must be taken into account. In fact, the best RSE code is the one which minimizes the probability that the desired QoS parameter exceeds the fixed threshold of the session. For example, if we take the first radio QoS class defined in our proposed wireless access system, we have to choose a coding scheme which minimizes the probability that a packet undergoes a delay higher than the fixed delay requirement of the class.

The obtained results motivate the use of adaptive FEC schemes where the parameters of FEC vary dynamically according to the wireless channel state, number of receivers and the desired QoS. This is the subject of the Chapter 7. Finally our contributions in this chapter are the performance evaluations of different RSE codes in terms of bandwidth, average per-packet delay and loss rate for the BSC and the GE model [NB00].

Chapter 7

QoS-Based Adaptive Error Control

As we saw in Chapter 6, the use of FEC improves the performance in most cases. However, the choice of the coding scheme in a hybrid ARQ/FEC depends on several parameters. Wireless channels are highly affected by unpredictable factors such as cochannel interference, adjacent channel interference, propagation path loss, shadowing and multipath fading. These factors may cause a high degradation of the channel bit error rate and a high retransmission rate. On the other hand, even in good channel conditions, the retransmission rate increases enormously if there is a high number of receivers in a session. Hence, choosing a fixed coding scheme may cause the waste of bandwidth during the normal behavior of the channel since the redundant information is not required due to the low bit error rate of the channel. On the other hand, during the temporary degradation of the network, the amount of redundancy may not be sufficient for receivers to recover from transmission errors. Even with good channel conditions, if there is a high number of receivers, the redundancy level of a code may not be sufficient. Therefore, the use of adaptive coding schemes for wireless channels is an issue that has to be studied thoroughly.

An adaptive algorithm needs to estimate the channel conditions of all receivers listening to the same session in order to adjust its coding parameters dynamically based on an optimization criteria. Adaptive coding has already been proposed in different contexts. It has been proposed for real-time applications in order to cope with retransmission delays in Internet [TP00] [BT96] [BFPT99] as well as in wireless networks [HOK99] [QS00] [KJDM96] [LZ96]. It has also been proposed for multicast communications [GBS94] [RKT98]. Once again, we observe that all the adaptive coding schemes designed for multicast communications are based on a fixed environment. The other works have considered a wireless network but their adaptation scheme is designed for a point-to-point communication mode. Our proposed approach is different from other adaptive algorithms since it is capable to adapt itself not only to the channel conditions but also to the number of receivers. It is based on a predictive mechanism in the sense that it forwards a certain number of redundant packets in the network before their necessity. It attempts to decrease the used bandwidth as much as possible while maintaining the desired QoS parameters.

In Chapter 6, we used a two state Markov model to capture the error sequences of a wireless channel. The model consisted of a Good state and a Bad state. In the Good state, errors occurred with a low probability while in the Bad state they occurred with a high

probability. In some cases, modeling a radio communication channel as a two-state Gilbert-Elliot channel is not adequate when the channel quality varies dramatically. In this case, a straightforward solution is to form a channel model with more than two states as in [WM95] [Fri67]. The advantage of using a Markov model for the radio channel lies on its facility to capture the burstiness of the error process as well as to predict the future states of the channel based on its present state. Prediction is useful due to the memory that exists in the physical channel. Our proposed scheme tries to take advantage of the channel memory in order to obtain better performance.

This chapter presents our adaptive algorithm. We start by the description of the adaptive coding mechanism. Then, we present the finite state Markov model used in this chapter. We extend our performance evaluations from a two state Markov model to the general case of a finite state Markov model. Our proposed prediction method as well as our adaptation policy are presented afterwards. Finally, we present some simulation results comparing the performance of our adaptive error control protocol with other protocols.

7.1 Adaptive Coding

We consider an *original code* $RSE(N_{max}, K_{max})$ with $H_{max} = N_{max} - K_{max}$. Using the shortening technique, we can derive a *basic code* $RSE(N, K)$ with the same number of redundant packets $H = H_{max} = N - K$. From this basic code, we can create a large set of RSE codes $RSE(n, k)$ with $k \leq K$ and $h \leq H$ using the shortening and puncturing techniques described in Chapter 6. The software coder proposed by Rizzo can be easily extended to support multiple block sizes and multiple redundant packets as in [Rub98]. The only implication of such a coder is that it needs to support the maximum data block size K_{max} which is normally bigger than the actual data block size k. However, taking the maximum data block size allows us to use a single generator matrix that can support up to K_{max} data packets which is important if we need to vary our coding parameters.

It is important to note that having a $K_{max} \geq k$ leads to a higher space complexity of the software coder. However, the time complexity is unaffected by the value K_{max}. The time complexity is only affected by the actual data block size k used in encoding/decoding. Therefore, choosing a high K_{max} does not affect the encoding and decoding speeds. In this chapter, we take an original code $RSE(255, 235)$ and a basic code $RSE(70, 50)$. Using this basic code, we can vary the coding parameters such that for any used code $RSE(n, k)$, we have $k \leq 50$ and $h \leq 20$.

7.2 Finite State Markov Model

We take a finite state Markov model as in [WM95]. This model is depicted in Figure 7.1. As it can be seen, the channel states associated with consecutive symbols are assumed to be neighboring states. This assumption is true for a slow fading channel where the SNR varies slowly compared to the symbol interval T.

Let $0 = \lambda_0 < \lambda_1 < \lambda_2 < ... < \lambda_S = \infty$ be the thresholds of the received SNR. The chan-

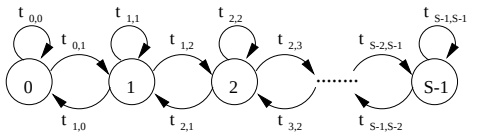


Figure 7.1: Finite state Markov model

nel is said to be in state s where $s \in \{0, 1, 2, ..., S-1\}$ if the received SNR is in the interval $[\lambda_s, \lambda_{s+1})$. Associated with each state, there is a binary symmetric channel with the error probability e_s. Recall that Rayleigh fading results in an exponentially distributed distortion of the signal. The probability density function of the SNR, $f(\lambda)$, follows an exponential distribution as in equation (6.18). Assuming that the channel fades slowly with respect to the symbol interval, T, the Markov transition probabilities can be approximated using the level crossing rate and the SNR density function as follows:

$$t_{s,s+1} \approx \frac{1}{\pi_s} exp(-\frac{\lambda_{s+1}}{\bar{\lambda}}) f_d T \sqrt{\frac{2\pi\lambda_{s+1}}{\bar{\lambda}}} \tag{7.1}$$

$$t_{s,s-1} \approx \frac{1}{\pi_s} exp(-\frac{\lambda_s}{\bar{\lambda}}) f_d T \sqrt{\frac{2\pi\lambda_s}{\bar{\lambda}}} \tag{7.2}$$

$$t_{s,s} = 1 - t_{s,s-1} - t_{s,s+1} \tag{7.3}$$

$$t_{0,0} = 1 - t_{0,1} \tag{7.4}$$

$$t_{S-1,S-1} = 1 - t_{S-1,S-2} \tag{7.5}$$

where f_d is the maximum doppler frequency and $\bar{\lambda}$ is the average SNR. The steady state probabilities, π_s, are:

$$\pi_s = \int_{\lambda_s}^{\lambda_{s+1}} f(\lambda) d\lambda = exp(\frac{-\lambda_s}{\bar{\lambda}}) - exp(\frac{-\lambda_{s+1}}{\bar{\lambda}}) \tag{7.6}$$

The error probabilities of each state e_s can be related to the received SNR according to the modulation scheme used in the system.

$$e_s = \frac{\int_{\lambda_s}^{\lambda s+1} f(\lambda) e_m(\lambda) d\lambda}{\int_{\lambda_s}^{\lambda s+1} f(\lambda) d\lambda} = \frac{1}{\pi_s} \int_{\lambda_s}^{\lambda s+1} f(\lambda) e_m(\lambda) d\lambda \tag{7.7}$$

where $e_m(\lambda)$ is the modulation function relating bit error probability to received SNR. The average error rate of the model can be found as $e = \sum_{s=0}^{S-1} \pi_s e_s$.

7.3 Performance Evaluation of FEC in a Finite State Markov Model

We consider a finite state Markov model as described in the previous section and we suppose that the the loss events at different receivers are independent. Once again, we assume that the channel is constant during a packet interval T. With this assumption, the packet loss probability of each state, p_s, can be calculated as in a BSC model with the error probability e_s. Without error correction coding, for a packet of length L bits, we have,

$$p_s = 1 - (1 - e_s)^L \tag{7.8}$$

Let us first consider the scenario where a sender multicasts data to R receivers using an ARQ scheme. The sender retransmits the original packet if there is at least one receiver that has not received the packet correctly. The sender sends k packets at a time before waiting for a feedback like protocol P1. The probability that a receiver loses a packet exactly l times is:

$$P(L_r = l) = \sum_{s=0}^{S-1} P_s(L_r = l), \tag{7.9}$$

$$P_s(L_r = l) = \begin{cases} \sum_{i=0}^{k-1} \Big[P_s(i, k-1) t_{s,s}(1 - p_s) \\ +P_{s-1}(i, k-1) t_{s-1,s}(1 - p_s) \\ +P_{s+1}(i, k-1) t_{s+1,s}(1 - p_s) \Big] & l = 0 \\ \sum_{i=0}^{k-1} \Big[P_s(i, k-1) t_{s,s} p_s \\ +P_{s-1}(i, k-1) t_{s-1,s} p_s \\ +P_{s+1}(i, k-1) t_{s+1,s} p_s \Big] & l = 1, 2, ... \end{cases}$$

$P_s(L_r = l)$ is the probability that a receiver loses a packet l times with the channel ending in state s. Like a two state model, we assumed that the channel remains at the same state during the time spanning the end of the transmission of a block and the beginning of the transmission of the next block. $P_s(i, k-1)$ represents the probability to have i packet losses in $k-1$ packet transmissions with the channel ending in state s. Extending the equation (6.34) to a finite state Markov model, the probability to have i packet losses in j packet transmissions is:

$$P(i, j) = \sum_{s=0}^{S-1} P_s(i, j), \qquad \text{for } i = 0, 1, 2, ..., s \text{ and } j = 1, 2, 3... \tag{7.10}$$

$$\begin{aligned} P_s(i,j) = & P_s(i,j-1)t_{s,s}(1-p_s) \\ & + P_{s-1}(i,j-1)t_{s-1,s}(1-p_s) \\ & + P_{s+1}(i,j-1)t_{s+1,s}(1-p_s) \\ & + P_s(i-1,j-1)t_{s,s}p_s \\ & + P_{s-1}(i-1,j-1)t_{s-1,s}p_s \\ & + P_{s+1}(i-1,j-1)t_{s+1,s} \end{aligned}$$

where $P_s(i,j)$ is the probability to have i packet losses in j packet transmissions with the channel ending in state s. In order to calculate $P_s(L_r = l)$, we set the initial conditions for the above recursions as $P_s(i,0) = 0$ for $i \neq 0$ and:

$$P_s(0,0) = \begin{cases} \pi_s & l = 0,1 \\ P_s(L_r = l-1) & l = 2,3,... \end{cases}$$

The efficiency of the scheme, Eff, can be calculated from equation (6.33) using equation (7.9) to find $P(L_r = l)$. Now, we consider protocol P2 where the sender uses an RSE code with a coded block size of n packets containing k original packets and h redundant packets. In this case, the sender sends k original packets followed by h redundant ones. Each receiver can recover from loss if it receives correctly k packets out of the $n = k + h$ transmitted packets. If the receiver can not recover from loss, it asks for a retransmission.

We define $Q(L_r = l)$ as the probability that a receiver loses a packet exactly l times in the case of FEC. $Q(L_r = l)$ is again the sum of $Q_s(L_r = l)$, the probability of a receiver to lose a packet exactly l times with the channel ending in state s.

$$Q(L_r = l) = \sum_{s=0}^{S-1} Q_s(L_r = l), \tag{7.11}$$

$$Q_s(L_r = l) = \begin{cases} \sum_{i=0}^{h-1} \Big[P_s(i,n-1)t_{s,s}p_s \\ + P_{s-1}(i,n-1)t_{s-1,s}p_s \\ + P_{s+1}(i,n-1)t_{s+1,s}p_s \Big] + \\ \sum_{i=0}^{n-1} \Big[P_s(i,n-1)t_{s,s}(1-p_s) \\ + P_{s-1}(i,n-1)t_{s-1,s}(1-p_s) \\ + P_{s+1}(i,n-1)t_{s+1,s}(1-p_s) \Big] & l = 0 \\ \sum_{i=h}^{n-1} \Big[P_s(i,n-1)t_{s,s}p_s \\ + P_{s-1}(i,n-1)t_{s-1,s}p_s \\ + P_{s+1}(i,n-1)t_{s+1,s}p_s \Big] & l = 1,2,... \end{cases}$$

The initial values for $P(i,j)$ in case of FEC are:

$$P_s(0,0) = \begin{cases} \pi_s & l = 0,1 \\ Q_s(L_r = l-1) & l = 2,3,... \end{cases}$$

and $Q_s(i,0) = 0$ for $i \neq 0$. Once $Q(L_r = l)$ is known, we can calculate the efficiency from equation (6.36).

Note that in all the above formulas we have $t_{s-1,s} = 0$ for $s = 0$ and $t_{s,s+1} = 0$ for $s = S-1$. Packet loss rate and delay of each scheme can be calculated by the same equations derived in Chapter 6 for a GE model using $P(i,j)$, $P(L_r = l)$ and $Q(L_r = l)$ of a finite state Markov model instead.

7.4 Prediction Method

Up to now, we have always considered steady state conditions for our performance evaluations. While each receiver experiences the same average bit error rate, its instantaneous bit error rate may be different from other receivers. In order for our adaptive algorithm to change its coding parameters dynamically, it must be able to predict the performance of each of the available error control schemes for the next block before actually transmitting it. We assume that the adaptive algorithm is informed about the channel state of all the receivers at the beginning of the transmission of each block. Once the channel state of all receivers at instant t is known, the algorithm can predict the evolution of channel conditions of the receivers for the next block taking advantage of the fact that the future states of the Markov chain depends only on its present state.

Let's consider protocol P1 first. In order to estimate its efficiency, we have to estimate $P(L_r = l)$ first. In the previous section, we used the steady state probabilities π_s as our initial values in order to calculate $P(i,j)$ in equation (7.9). Assuming that a receiver is in state s' at the beginning of the transmission, the initial conditions for $P(i,j)$ in equation (7.9) are:

$$P_s(0,0) = \begin{cases} 1 \quad \text{if} \quad s = s' & l = 0,1 \\ 0 \quad \text{otherwise} & l = 0,1 \\ P_s(L_r = l-1) & l = 2,... \end{cases}$$

Using the above initial conditions, we get S different values for $P(i,j)$ and $P(L_r = l)$ depending on the state where the receiver was at the beginning of the transmission. We represent these probabilities by $P(L_r = l|s')$ and $P(i,j|s')$ where s' is the state of a receiver at the beginning of the transmission. We represent the number of receivers in each of the states of the Markov chain by $\{r_0, r_1, .., r_{S-1}\}$. It is clear that we have $\sum_{s=0}^{S-1} r_s = R$. The algorithm estimates the efficiency of P1 for R receivers as follows:

$$Eff = \frac{1}{E[M]} = \frac{1}{\sum_{m=1}^{\infty}\left(1 - \prod_{s'=0}^{S-1}(1 - P(L_r = m-1|s'))^{r_{s'}}\right)} \tag{7.12}$$

Now, we consider protocol P2. In order to estimate the efficiency of protocol P2, the adaptive algorithm needs to estimate $Q(L_r = l)$ first. Assuming that a receiver is in state s' at the beginning of a transmission, the initial conditions for $P(i,j)$ in equation (7.11) are:

$$P_s(0,0) = \begin{cases} 1 \quad \text{if} \quad s = s' & l = 0,1 \\ 0 \quad \text{otherwise} & l = 0,1 \\ Q_s(L_r = l-1) & l = 2,... \end{cases}$$

Note that once again, we have different $Q(L_r = l)$ probabilities depending on the channel state at the beginning of the transmission. We represent these probabilities by $Q(L_r = l|s')$ where s' is the channel state of a receiver at the beginning of the transmission. The algorithm predicts the efficiency of protocol P2 as follows:

$$Eff = \frac{1}{E[M]} = \frac{k}{n}\frac{1}{\sum_{m=1}^{\infty}\left(1 - \prod_{s'=0}^{S-1}(1 - Q(L_r = m-1|s'))^{r_{s'}}\right)} \tag{7.13}$$

The next QoS metric is the packet loss rate which is the probability to have at least one receiver that has not received a packet correctly after the first transmission. Considering protocol P1, the probability to receive a packet correctly after the first transmission is $P(L_r = 0)$. Once again, we define r_s as the number of receivers in state s. $P(L_r = 0|s')$ is the probability of a receiver to receive a packet correctly after the first transmission with the receiver being in state s' at the beginning of the transmission. The packet loss rate of protocol P1 is estimated as follows:

$$PLR = 1 - \prod_{s'=0}^{S-1}\Big[P(L_r = 0|s')\Big]^{r_{s'}} \tag{7.14}$$

In case of protocol P2, the probability that a receiver gets a packet correctly after the first transmission is $Q(L_r = 0)$. We represent this probability with the receiver being in state s' at the beginning of the transmission by $Q(L_r = l|s')$. The adaptive algorithm estimates the packet loss rate of protocol P2 as below:

$$PLR = 1 - \prod_{s'=0}^{S-1}\Big[Q(L_r = 0|s')\Big]^{r_{s'}} \tag{7.15}$$

The last metric that we have to consider is the average per-packet delay. Let us begin by protocol P1. In order to estimate the average delay of a packet, we need to estimate the average number of transmissions for a packet first. This parameter has already been estimated in equation (7.12). Moreover, we have to predict the probabilities of having R NAKs and no NAK for a block of k packets. These probabilities can be found below. Remember that the probability to have a NAKs is the probability to have a receivers that have lost at least one packet in the transmitted block.

$$\Phi(A = R) = \prod_{s'=0}^{S-1} \Big[1 - P(0, k|s')\Big]^{r_{s'}} \tag{7.16}$$

$$\Phi(A = 0) = \prod_{s'=0}^{S-1} \Big[P(0, k|s')\Big]^{r_{s'}} \tag{7.17}$$

$P(0, k|s')$ is the probability to have zero losses in k transmissions with the receiver in state s' at the beginning of the transmission. This probability can be calculated from equation (7.10) using the following initial values. The average per-packet delay can be estimated from equation (6.11) by using equations (7.12), (7.16) and (7.17).

$$P_s(0, 0) = \begin{cases} 1 & \text{if } \ s = s' \\ 0 & \text{otherwise} \end{cases}$$

Now let us consider protocol P2. For this protocol, the adaptive algorithm needs to estimate the probability of decoding as well as the probabilities to have R NAKs and no NAK before estimating the average per-packer delay. It estimates the probability of decoding P_d as:

$$P_d = \prod_{s'=0}^{S-1} \Big[Q(L_r = 0|s')\Big]^{r_{s'}} - \prod_{s'=0}^{S-1} \Big[P(L_r = 0|s')\Big]^{r_{s'}} \tag{7.18}$$

The probabilities to have R NAKs and no NAK in case of P2 are estimated as follows:

$$\Phi(A = R) = \prod_{s'=0}^{S-1} \Big[1 - \sum_{i=0}^{h} P(i, n|s')\Big]^{r_{s'}} \tag{7.19}$$

$$\Phi(A = 0) = \prod_{s'=0}^{S-1} \Big[\sum_{i=0}^{h} P(i, n|s')\Big]^{r_{s'}} \tag{7.20}$$

where again $P(i, n|s')$ represents the probability to have i losses in n transmissions with th receiver in state s' at the beginning of the transmission. The initial values are the same as equations (7.16) and (7.17). Finally, the average delay of a packet for protocol P2 can be calculated from equation (6.17) using equations (7.13), (7.18), (7.19) and (7.20).

7.5 Adaptation Policy

Let $C = \{c_0, c_1, ..., c_k\}$ be the set of RSE codes available at the sender. The sender can either choose the ARQ/FEC error control protocol with an RSE code in C or a pure ARQ protocol. According to the variations of SNR, the receiver channel may be in one of the states of the Markov model at each instant t. We assume that the sender knows the state of the Markov chain at the transmission time for all receivers. Let's define the *transmission status* at time t as the set of all tuples (s, r_s) where $s \in \{0, 1, ..., S-1\}$ is the channel state in the Markov model and r_s is the number of wireless receivers in state s at time t.

Before transmitting, the adaptive algorithm in the sender must estimate the efficiency, packet loss rate and delay of the ARQ/FEC protocol using all the available coding schemes as well as the ARQ protocol as a function of the transmission status. It then tries to find the protocol satisfying the desired packet loss rate and delay. If there are several protocols satisfying these criteria, the algorithm must choose the one with the highest efficiency. Note that our adaptive approach is predictive rather than reactive since the sender tries to predict the channel conditions as well as the evolution of QoS metrics for all receivers before actually sending a block. The sender then chooses a protocol according to its predictions.

The time is divided into transmission rounds. Each transmission round corresponds to the transmission of n packets in case of FEC and k packets in case of ARQ. A transmission round ends when the sender is informed about the reception states of all receivers. The adaptive algorithm is repeated at the end of each transmission round. Basically, the algorithm goes through the following steps:

1. At the beginning of the algorithm, the sender determines the desired packet loss rate and delay of the session. It also determines the transmission status.
2. The sender estimates the packet loss rate and the delay of the ARQ protocol as well as the ARQ/FEC protocol using all the available coding schemes, based on the transmission status. If it finds several protocols satisfying the QoS metrics of the session, it chooses the one with the highest efficiency. It then adjusts its parameters and starts the transmission of the block.
3. At the end of a transmission round, the sender again determines the transmission status. It then repeats the step 2.

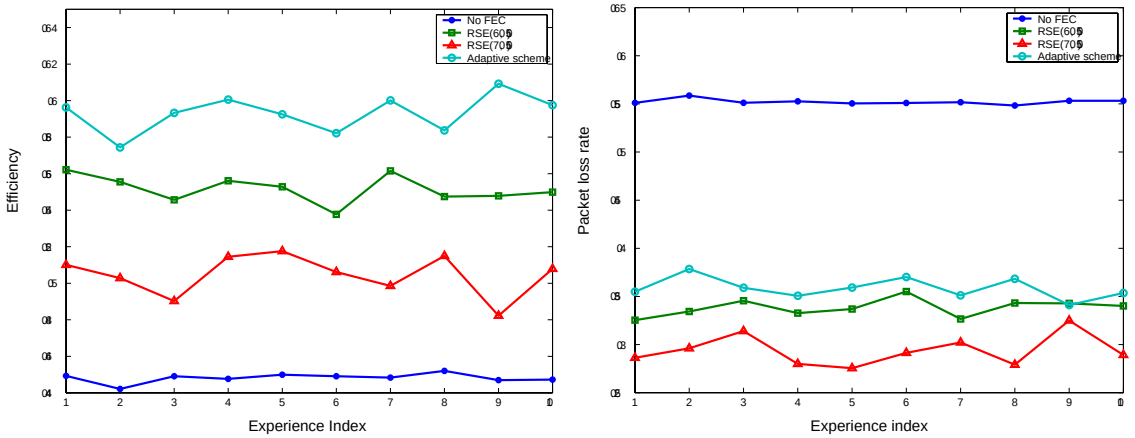
In the above algorithm, the sender finds the best error control mechanism in real-time right before transmitting a block. One possible optimization is for the sender to make a table of optimal mechanisms called FEC_TABLE indexed on transmission status. Each time, the sender estimates the QoS metrics for a transmission status, it adds an entry to the FEC_TABLE. In this way, if a transmission status occurs again, the sender can find the best mechanism by a simple table lookup. If the service differentiation is limited to some classes with predefined QoS metrics such as our proposed wireless access system with three level of service differentiation, then a table can be generated for each QoS class off-line.

One may argue that the sender risks to consume a lot of memory if there is a high number of receivers or if the Markov model used to model the wireless channel has many states.

It is clear that the choice between using a precomputed table or an online estimation is a tradeoff between the consumed memory and the complexity of the algorithm. However, the simulation results show that the choice of the best error control protocol does not vary significantly for a wide range of transmission status.

7.6 Simulation Results

We have carried out several simulations in OPNET which is an event-driven simulation tool. The date rate is 20 Mb/s and the carrier frequency is 5.2 GHz. As before, the data and control packets have 54 and 9 bytes respectively. We use a BPSK modulation scheme. The average SNR is 34 dB corresponding to an average bit error probability of 10^{-4}. All the receivers are located within a distance of 23 meters from the AP. The wireless channel is modeled by a 3 state Markov model in the OPNET environment. The state s_0 of the Markov model corresponds to a Bad state with $e_0 \approx 1$, the state s_1 corresponds to an intermediate state with a non-zero error probability $e_1 \approx 2 \times 10^{-5}$ and the state s_2 corresponds to a Good state with a zero error probability $e_2 \approx 0$. In order to have an error probability of $e_0 \approx 1$, λ_1 must be equal to 2dB in BPSK. For a zero error probability $e_2 \approx 0$ in state s_2, we also need to fix λ_2 at 34dB in BPSK. Knowing the threshold values of the Markov model, all the other parameters can be easily found as in Section 7.2.



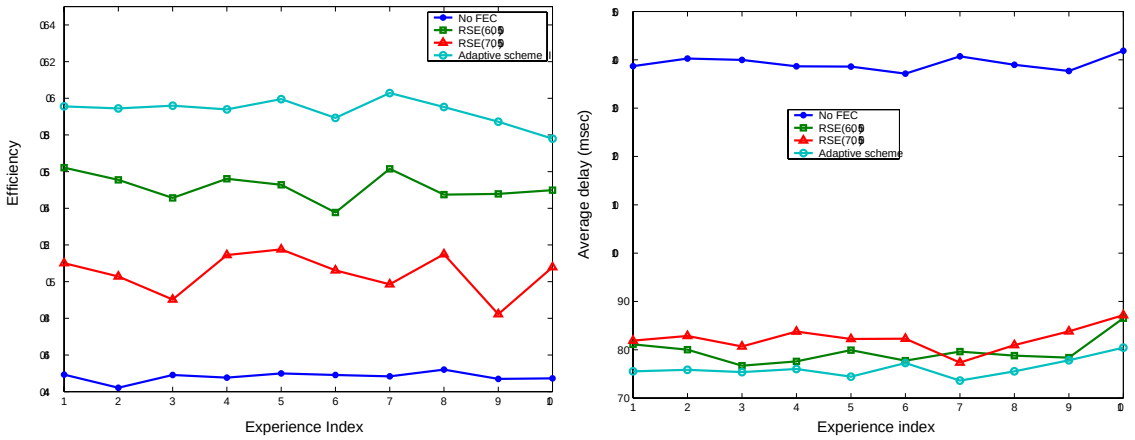
(a) Efficiency

(b) Packet loss rate

Figure 7.2: Simulation results for $PLR = 50\%$

For each scenario, we have carried out ten different simulations, each with a different seed. Figure 7.2 compares the efficiency and the packet loss rate of our proposed adaptive scheme with a pure ARQ protocol, an ARQ/FEC protocol using $RSE(60, 50)$ and another hybrid protocol using $RSE(70, 50)$. The number of receivers is fixed at 1000. This figure corresponds to a scenario where non-real-time traffic such as data is transmitted over the network. The protocols try to retransmit a packet until it is correctly received by all receivers. We have chosen a $PLR = 50\%$ in order to reduce the retransmission rate by a half in case

of adaptive scheme. From this figure, we can observe that the adaptive scheme provides the best efficiency. It also has a PLR less than 50% as it was expected. Although other fixed hybrid protocols provide better packet loss rates, they have a lower efficiency compared to our adaptive scheme.



(a) Efficiency

(b) Average delay

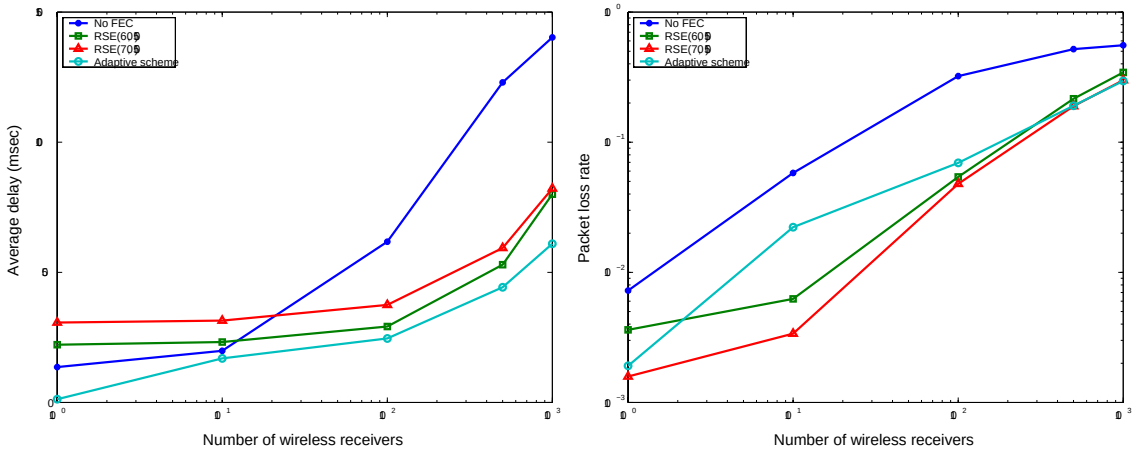
Figure 7.3: Simulation results for a packet life time of 100 msec

Figure 7.3 compares the efficiency and the average delay of our proposed adaptive scheme with the same error control mechanisms as in the previous figure. The number of receivers is again 1000. This scenario corresponds to a traffic with delay constraints. The maximum lifetime of each packet of this traffic is fixed at 100 msec. We observe that our proposed adaptive scheme provides the lowest average delay while maximizing the efficiency. The other hybrid ARQ/FEC protocols also provide an average delay lower than 100 msec but they have a lower efficiency than our protocol.

Figure 7.4 depicts the average delay and the packet loss rate of our adaptive algorithm and the same error control protocols as before as a function of the number of receivers. The packet life time is fixed at 50 msec and the packet loss rate is 10%. Our adaptive mechanism can guarantee an average delay of 50 msec up to 600 receivers while the ARQ mechanism, for example, can guarantee this delay up to 30 receivers as it can be seen in the first plot. The packet loss rate is also less than 10% up to 200 receivers in our adaptive scheme and less than 20 receivers in the case of ARQ protocol.

7.7 Conclusion

In this chapter, we proposed an adaptive algorithm capable to switch between an ARQ and a set of ARQ/FEC error control protocols. The coding scheme used in the ARQ/FEC protocols is based on RSE codes. The adaptive algorithm chooses the best error control mechanism as a function of the channel bit error rate, the channel state of the receivers and the desired QoS metric of the session while maximizing efficiency. We used a finite state Markov chain as



(a) Average delay

(b) Packet loss rate

Figure 7.4: Simulation results for a packet life time of 50 msec and a PLR of 10%

our wireless channel model. This model allowed us to predict the future states of the channel for each receiver based on its current channel state. Simulation results showed that the use of FEC improves significantly the performance of error control mechanisms specially when there is a high number of receivers listening to the same session. It also showed that the use of adaptive mechanism is quite useful in order to save bandwidth while maintaining the QoS metrics of a session below their thresholds.

Several issues need more investigation:

- What is the effect of the feedback losses? During our analysis and simulations, we assumed that no control message is lost.
- What is the effect of other QoS metrics? We considered efficiency, packet loss rate and average delay for our analysis. Other QoS metrics such as jitter, dropping rate and power consumption of mobile terminals must also be considered.
- How is the channel state information of the receivers available at the sender? During our analysis and simulations, we assumed that the sender is informed about the channel state of all the receivers. However, we did not specify the exact mechanism. This information may be available at the sender either by the receivers informing the sender about their SNR or by the sender itself making an estimation of the number of receivers at each state. In the first case, the format of NAK messages can be modified in order to include the receiver SNR at the end of the reception of a block. In case a receiver has received all the packets correctly, it has to send an extra control message to the sender in order to inform it about its channel state. In the second case, the sender may estimate the channel state of all receivers based on their number of lost packets in a block.
- How does our proposed adaptive algorithm perform in a real wireless channel model? Within the WAND project, a *Stochastic Radio Channel Model* (SRCM) for broadband

indoor mobile systems operating at 5.2 GHz has been derived [HBF97]. The use of SRCM as our wireless channel model in our simulations will allow us to make an efficient and realistic evaluation of our adaptive algorithm.

Finally our contributions in this chapter are the proposal of an adaptive coding scheme [NLB00] and the simulations comparing the performance of our adaptive scheme with other schemes in terms of packet loss rate, average delay and efficiency.

Chapter 8

Conclusions and Future Work

This chapter concludes this dissertation with a summary of our contributions and possible directions for future work.

8.1 Summary

The first result of this dissertation was the design of a wireless IP-based access network which enabled the transmission of IP traffic directly over the radio channel. Chapter 2 provided the general architecture of the system as well as its functional entities. During the design of the system, we identified three fundamental challenges:

- QoS support
- Mobility support
- Multicast support

In Chapter 3, we presented an architecture for QoS support in the access network. In this architecture, the QoS scheme of the access network is completely independent of the backbone QoS scheme. In the access network, the network level QoS parameters are mapped to the radio QoS classes. The incoming traffic is classified into three general radio QoS classes based on its QoS information. Service differentiation in the radio access network is done using different queuing and reliability mechanisms at the radio link layer.

In Chapter 4, we presented a framework for mobility management used in the access network. Intra-subnet handover is handled by the radio link layer while inter-subnet handover is based on the IPv6 mobility management mechanism. In intra-subnet handover, an MT moves to another AP but it stays within the same IP domain. This kind of handover can be handled quite fast since all the traffic characteristics of the MT is already available at the router. Inter-subnet handover, however, is more complicated since the MT changes its IP domain and as a consequence obtains a new address. In this case, the new router must be informed about the traffic characteristics of the MT before transmitting them over its local

network. The mobility management scheme offered by mobile IPv6 is appropriate for *macro* mobility where the MT does not move its point of attachment to the Internet frequently. Furthermore, these handovers may be lossy to the time that it takes for the CNs to be informed about the MT's new address. In order to reduce the losses during an inter-subnet handover, the router of the network in which the MT was located before handover intercepts the traffic destined to the MT and forwards it to the new location of the MT.

Another important issue in case of inter-subnet handover is the QoS support problem. The new MR must be rapidly informed about the QoS requirements of the MT. Furthermore, due to the change of the MT's address, the reservations made in the backbone routers along the path from the MT to its CNs must be updated in order to reflect the MT's new address. This problem only arises if the backbone QoS mechanism is based on the integrated services architecture.

In Chapter 5, we presented a complete framework for multicast communication in the access network. We proposed a new group management protocol in order to track the group members in the access network. The numerical results showed that this protocol is more efficient than the IGMP in terms of bandwidth use. We also proposed to forward the multicast traffic only to the APs that have active members of the group to which the traffic is destined. The proposed multicast framework is complemented by a link layer addressing scheme used to address groups in each AP. The AP allocates this address to a group if it has at least one member of the group in its cell. It also communicates this address to the group members in its coverage area. This address is unique in each AP.

By the end of the Chapter 5, the design of the wireless IP access system was finished, though lots of open issues still remained. From all these research perspectives, we chose to investigate the QoS problem. More precisely, our research was focused on the service differentiation at the wireless data link control layer. The result of this part was the proposition of an adaptive error control mechanism as a way to control QoS at the radio access network. The main problems that we treated in this part were:

- The effect of different error control mechanism on QoS
- The effect of different number of receivers on QoS

In Chapter 6, we presented several numerical results comparing the performance of different error control mechanisms and different coding parameters. The performance evaluations were done based on three main QoS metrics: bandwidth use, packet loss rate and average delay. We used Reed-Solomon erasure codes. Several numerical analyses were done for different wireless channel models, different bit error rates, different number of wireless receivers and different coding parameters for each of our specified QoS metrics. These results showed that the use of FEC causes an improvement in the performance of the error control mechanism in most cases. Nevertheless, choosing a code that can perform efficiently in all channel conditions and for any number of wireless receivers is a difficult task. Therefore, an efficient error control protocol must be able to change its coding parameters dynamically.

In Chapter 7, we proposed an adaptive QoS-based error control mechanism. The protocol uses a finite state Markov model in order to predict the evolution of the channel conditions

of each receiver. Based on this prediction, the protocol changes its error control strategy as well as its coding parameters for the RSE codes. Simulation results compared the performance of our proposed adaptive protocol with fixed error control protocols. In general, we observed that our adaptive protocol reduces the bandwidth use of the network compared to other schemes, while respecting the QoS requirements of the receivers.

8.2 Future Directions

QoS support is still one of the hot research topics in the fixed Internet. Wireless links add more constraints to the QoS support due to their low bandwidth and high bit error rates compared to wired links. Several research perspectives exist in the area of QoS control for access networks. We just cite some of the most important issues that are complementary to our work.

Our proposed adaptive protocol took bandwidth, packet loss rate and average delay as its QoS parameters. One possible direction for future work is to consider other QoS parameters such as jitter, dropping rate and the power consumption of the terminals. For our performance evaluations, we assumed that no control packets are lost. The effect of control packet losses on the overall performance of the protocols is an interesting subject to study. Finally, it is also interesting to evaluate the performance of our adaptive error control protocol over real wireless systems such as the SRCM developed during the WAND project.

During the design of the access network, we proposed the use of a scheduling entity that controls the flow of traffic in the access network taking into account their QoS requirements. We believe that in this case, a narrow collaboration of the scheduling entity and the wireless link layer control is the only way to assure an efficient service differentiation at the access network. The effect of a QoS-based error control mechanism on the scheduling algorithm is one of the important directions for our future work.

Another open issue is the effect of FEC on the congestion. During congestion, packets are lost due to buffer overflows. First of all, the adaptive algorithm must distinguish the losses due to congestion from other types of losses. Increasing the redundancy level of coding beyond a certain limit may increase the congestion state of the access network. As a result, the degree of redundancy injected to the access network must be adjusted according to the congestion state of the network. Therefore, it is important to evaluate the performance of FEC as a function of the traffic load of the system.

Bibliography

[AB96] A. Acharya and B. R. Badrinath. A framework for delivering multicast messages in networks with mobile hosts. *ACM/Baltzer Journal of Mobile Networks and Applications*, 1(2), 1996.

[AHK+96] P. Agrawal, E. Hyden, P. Krzyzanowski, P. Misthra, M. Srivastava, and J. Trotter. Swan: A mobile multimedia wireless network. *IEEE Personal Communications*, April 1996.

[ALRR97] A. Acharya, J. Lin, B. Rajagopalan, and D. Raychaydhuri. Mobility management in wireless ATM networks. *IEEE Communications Magazine*, 35(11):100–109, November 1997.

[ALSNS98] Juha Ala-Laurila, Lorraine Stacey, Neda Nikaein, and Jukka Seppala. Designing a Wireless Broadband IP System with QoS Guarantees. In *Proceedings of ACTS Mobile Summit '98*, June 1998.

[APP+95] E. Ayanoglu, S. Paul, T. F. La Porta, K. K. Sabnani, and R. D. Gitlin. AIRMAIL: A link-layer protocol for wireless networks. *ACM/Baltzar Wireless Networks*, 1(1), January 1995.

[ATM97] ATM Forum. LAN emulation over ATM specification - version 2.0, July 1997.

[Bal97] A. Ballardie. Core based trees (CBT version 2) multicast routing - protocol specification. RFC 2189, September 1997.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. RFC 2475, December 1998.

[BCP00] J. Bound, M. Carney, and C. Perkins. Dynamic host configuration protocol for IPv6 (DHCPv6). Internet Draft, draft-ietf-dhc-dhcpv6-15.txt, May 2000.

[BFPT99] Jean-Chrysostome Bolot, Sacha Fosse-Parisis, and Don Towsley. Adaptive FEC-Based error control for interactive audio in the internet. In *proceedings of INFOCOM'99*, New York, March 1999.

[BRA98] BRAN group. HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional specification. ETSI Standard, July 1998.

[BRA00] BRAN group. HIgh PErformance Radio Local Area Network (HIPERLAN) Type 2; System Overview. ETSI Standard, August 2000.

[BT96] Jean Bolot and Thierry Turletti. Adaptive error control for packet video in the internet. In *proccedings of ICIP'96*, Lausanne, Switzerland, September 1996.

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reservation protocol (RSVP) - version 1 functional specification. RFC 2205, September 1997.

[CD98] A. Conta and S. Deering. Generic packet tunneling in IPv6 specification. RFC 2473, December 1998.

[DDC97] C. Diot, W. Dabbous, and J. Crowcroft. Multipoint communications: a survey of protocols, functions and mechanism. *IEEE Journal on selected areas in communications (JSAC)*, 15(3):277–290, April 1997.

[Dee89] S. Deering. Host extensions for IP multicasting. RFC 1112, August 1989.

[DH98] S. Deering and R. Hinden. Internet protocol, version 6 (IPv6) specification. RFC 2460, December 1998.

[DNP99] M. Degermark, B. Nordgren, and S. Pink. IP header compression. RFC 2507, February 1999.

[DR99] C. Dovrolis and P. Ramanathan. A case for relative differentiated services and the proportional differentiation model. *IEEE Network*, pages 26–34, September 1999.

[Ell63] E.O. Elliot. Estimation of error rates for codes on burst-error channels. *Bell Systems Technical Journal*, page 1977, September 1963.

[ES98] David Eckhardt and Peter Steenkiste. Improving wireless LAN performance via adaptive local error control. In *Proceedings of Sixth IEEE International Conference on Network Protocols ICNP'98*, Austin, October 1998.

[Fen97] W. Fenner. Internet group management protocol, version 2. RFC 2236, November 1997.

[FH98] P. Ferguson and G. Huston. *Quality of Service, Delivering QoS on the Internet and in Corporate Networks*. Wiley Computer Publishing, January 1998.

[FHNS98] G. Fankhauser, S. Hadjiefthymiades, N. Nikaein, and L. Stacey. RSVP support for mobile IP version 6 in wireless environments. Internet Draft, November 1998.

[FHNS99] G. Fankhauser, S. Hadjiefthymiades, N. Nikaein, and L. Stacey. Interworking RSVP and mobile IP version 6 in wireless environments. In *4th ACTS Mobile Communication Summit*, Sorrento, Italy, June 1999.

[FJL+97] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for ligth-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5:784–803, December 1997.

[Fri67] B. D. Fritchman. A binary channel characterization using partitioned Markov chains. *IEEE Transactions on Information Theory*, IT-13(2), April 1967.

[GBS94] P. Godlewski, M. Braneci, and A. Serhrouchni. An error control scheme for multicast communications over an ATM network. In *Proceedings of Singapore International Conference on Communication Systems (ICCS'94)*, Singapore, November 1994.

[GGB96] M. Gagnaire, P. Godlewski, and M. Braneci. An intelligent hybrid type II ARQ/FEC logical link control protocol for GSM mobile communication system. In *Proceedings of IEEE VTC*, Atlanta, USA, April 1996.

[Gil60] E.N. Gilbert. Capacity of a burst-noise channel. *Bell Systems Technical Journal*, page 1253, September 1960.

[HBF97] R. Heddergott, U. P. Bernhard, and B. H. Fleury. Stochastic radio channel model for advanced indoor mobile communication systems. In *Proceedings of the 8th IEEE Internationsl Symposium on Personal, Indoor and Mobile Radio Communications PIMRC'97*, volume 1, pages 140–144, Helsinki, Finland, September 1997.

[HOK99] Chi-Yuan Hsu, Antonio Ortega, and Masoud Khansari. Rate control for robust video transmission over burst-error wireless channels. *IEEE JSAC*, 17(5):756–773, May 1999.

[HPFM98] Stathes Hadjiefthymiades, Sarantis Paskalis, George Fankhauser, and Lazaros Merakos. Mobility management in an IP-based wireless ATM network. In *Proceedings of ACTS Mobile Summit '98*, June 1998.

[Hui96] Christian Huitema. The case for packet level FEC. In *Proceedings of IFIP 5th International Workshop on Protocols for High Speed Networks (PfHsn'96)*, INRIA, Sophia Antipolis, FRANCE, October 1996.

[IEE99] IEEE Standard. IEEE 802.11 wireless MAC and PHY specifications. Draft Version, 1999.

[Jef94] R. Jeffries. ATM LAN emulation, 1994.

[Kal90] S. Kallel. Analysis of a type II hybrid ARQ scheme with code combining. *IEEE Transactions on Communications*, 38(8), August 1990.

[KH90] S. Kallel and D. Haccoun. Generalized type II hybrid ARQ scheme using punctured convolutional coding. *IEEE Transactions on Communications*, 38(11):1938–1946, November 1990.

[KHM98] A. Kaloxylos, S. Hadjiefthymiades, and L. Merakos. Mobility management and control protocol for wireless ATM networks. *IEEE Network, special issue on PCS Network Management*, pages 19–27, July/August 1998.

[KJDM96] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein. Low bit-rate video transmission over fading channels for wireless microcellular systems,. *IEEE Transactions on Circuits and Systems for Video Technolog*, 6(1):1–11, 1996.

[KNE97a] Y. Katsube, K. Nagami, and H. Esaki. Internetworking based on cell switch router - architecture and protocol overview. *Proceedings of the IEEE*, 85(12):1998–2006, December 1997.

[KNE97b] Y. Katsube, K. Nagami, and H. Esaki. Toshiba's router architecture extensions for ATM: Overview. RFC 2098, February 1997.

[LC83] S. Lin and D. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[LGLA98] B. Levine and J.J. Garcia-Luna-Aceves. A comparison of reliable multicast protocols. *Multimedia Systems (ACM/Springer)*, 1998.

[LH98] M. Laubach and J. Halpern. Classical IP and ARP over ATM. RFC 2225, April 1998.

[LMZG97] H. Liu, H. Ma, M. Zarki, and S. Gupta. Error control schemes for networks: An overview. *Mobile Networks and Applications*, 2(2):167–182, October 1997.

[LY82] S. Lin and P.S. Yu. A hybrid ARQ scheme with parity retransmission for error control of satellite channels. *IEEE Transactions on Communications*, 30(7):1701–1719, July 1982.

[LZ96] H. Liu and M. El Zarki. Delay bounded type-II hybrid ARQ for video transmission over wireless networks. In *Proceedings of Conference on Information Sciences and Systems*, Princeton, NJ, March 1996.

[MAA+98] J. Mikkonen, J. Aldis, G. Awater, A. Lunn, and D. Hutchison. The Magic WAND-functional overview. *IEEE Journal on Selected Areas in Communications*, 16(6), August 1998.

[MB99] J. Meierhofer and U. P. Bernhard. Link quality control for wireless ATM networks. In *Proceedings of IEEE ICC'99*, pages 1874–1879, Vancouver, Canada, June 1999.

[McA90] Anthony J. McAuley. Reliable broadband communication using a burst erasure correcting code. In *Proceedings of SIGCOMM'90*, Philadelphia, Pennsylvania, September 1990.

[Mei98] J. Meierhofer. Data link control for indoor wireless ATM networks. In *Proceedings of Wireless '98*, pages 517–525, Calgary, Canada, July 1998.

[MJV96] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *Proceedings of SIGCOMM 96*, pages 117–130, Stanford, CA, August 1996.

[MLLV98] J. Mikkonen, L. Lehtinen, J. Lahti, and S. Veikkolainen. A radio access network architecture for IP QoS. In *Proceedings of MoMuC'98*, pages 109–119, Berlin, Germany, October 12-14 1998.

[Moy94] J. Moy. Multicast extensions to OSPF. RFC 1584, March 1994.

[NB98] N. Nikaein and C. Bonnet. Wireless multicasting in an IP environment. In *Proceedings of 5th International Workshop on Mobile Multimedia Communication IEEE MoMuc '98*, October 1998.

[NB99] Neda Nikaein and Christian Bonnet. Le multicast sans fil dans un environnement IP. In *Congrès DNAC (De Nouvelles Architectures pour les Communications)*, Paris, France, December 1999.

[NB00] N. Nikaein and C. Bonnet. On the performance of FEC for multicast communication on a fading channel. In *Proceedings of International Conference on Telecommunications IEEE ICT'00*, Acapulco, Mexico, May 2000.

[NBT98] J. Nonnenmacher, E. Biersack, and D. Towsley. Parity-based loss recovery for reliable multicast transmission. *IEEE/ACM Transactions on Networking*, 6(4):349–361, August 1998.

[NEH+96] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. Ipsilon flow management protocol specification for IPv4 version 1.0. RFC 1953, May 1996.

[NLB00] N. Nikaein, H. Labiod, and C. Bonnet. MA-FEC: a QoS-based adaptive FEC for multicast communication in wireless networks. In *Proceedings of International Conference on Communications IEEE ICC'00*, New Orleans, USA, June 2000.

[NLM96] P. Newman, T. Lyon, and G. Minshall. Flow labelled IP: a connectionless approach to ATM. In *Proceedings of INFOCOM'96*, March 1996.

[NML97] P. Newman, G. Minshall, and T. Lyon. IP switching: ATM under IP. *IEEE/ACM Transactions on Networking*, November 1997.

[NNS98] T. Narten, E. Nordmark, and W. Simpson. Neighbor discovery for IP version 6 (IPv6). RFC 2461, December 1998.

[Per98] Charles Perkins. *Mobile IP, Design, Principals and Practices*. Addison-Wesley, 1998.

[PJ96] C.E. Perkins and D.B. Johnson. Mobility support in IPv6. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom'96)*, November 1996.

[Pro89] J. G. Proakis. *Digital Communications*. New York: MacGrawhill, 2nd edition, 1989.

[PSA96] Sassan Pejhan, Mischa Schwartz, and Dimitris Anastassiou. Error control using retransmission schemes in multicast transport protocols for real-time media. *IEEE/ACM Transactions on Networking*, 4(3):413–427, June 1996.

[PSLB97] S. Paul, K. K. Sabnani, J. C.-H. Lin, and S. Bhattacharyya. Reliable multicast transport protocol. *IEEE Journal on selected areas in communications (JSAC)*, 15(3):407–421, April 1997.

[QS00] Daji Qiao and Kang G. Shin. A two-step adaptive error recovery scheme for video transmission over wireless networks. In *proceedings of INFOCOM'00*, Tel Aviv, Israel, March 2000.

[RDK+97] Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow. Cisco systems' tag switching architecture overview. RFC 2105, February 1997.

[RDR+97] Y. Rekhter, B. Davie, E. Rosen, G. Swallow, D. Farinacci, and D. Katz. Tag switching architecture overview. *Proceedings of the IEEE*, 85(12):1973–1983, December 1997.

[Riz97] Luigi Rizzo. On the feasibility of software fec. *Computer Communication Review*, April 1997.
Source code available at http://www.iet.unipi.it/ luigi/fec.html.

[Riz98] L. Rizzo. Fast group management in igmp. In *Proceedings of HIPPARCH'98 workshop*, June 1998.

[RKT98] Dan Rubenstein, Jim Kurose, and Don Towsley. Real-time reliable multicast using proactive forward error correction. In *NOSSDAV'98*, pages 279–293, Cambridge, England, July 1998.

[RS60] I.S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, June 1960.

[Rub98] Dan Rubenstein. Increasing the functionality and availability of reed-solomon fec codes: a performance study. Technical Report UMass CMPSCI Technical Report 98-31, University of Massachusetts, August 1998.

[SALM+98] L. Stacey, J. Ala-Laurila, J. Mikkonen, J. Seppälä, S. Hadjiefthymiades, N. Nikaein, and G. Fankhauser. IP over wireless ATM. ACTS Magic WAND public report, August 1998.

[SDH96] G. Sharma, A. Dholakia, and A. Hassan. Simulation of error trapping decoders on a fading channel. In *Proceedings of VTC*, 1996.

[SK94] H. Saran and S. Keshav. An empirical evaluation of virtual circuit holding times in IP over ATM networks. In *Proceedings of IEEE INFOCOM'94*, pages 1132–1140, June 1994.

[TBA97] A. Talukdar, B. Badrinath, and A. Acharya. MRSVP: A Resource Reservation Protocol for an Integrated Services Packet Network with Mobile Hosts. Technical report DCS-TR-337, Rutgers University, 1997.

[TKP97] D. Towsley, J. Kurose, and S. Pingali. A comparison of sender-initiated and receiver-initiated reliable multicast protocols. *IEEE Journal on selected areas in communications (JSAC)*, 15(3):398–406, 1997.

[TKWZ00] A. Terzis, J. Krawczyk, J. Wroclawski, and L. Zhang. RSVP operation over IP tunnels. RFC 2746, January 2000.

[TN98] S. Thomson and T. Narten. IPv6 stateless address autoconfiguration. RFC 2462, December 1998.

[TP00] Tsunyi Tuan and Kihong Park. Multiple time scale redundancy control for QoS-sensitive transport of real-time traffic. In *Proceedings of INFOCOM'00*, Tel Aviv, Israel, March 2000.

[VD97] M. Veeraraghavan and G. Dommety. Mobile location management in ATM networks. *IEEE JSAC*, October 1997.

[Wan94] H.S. Wang. On verifying the first-order Markovian assumption for a Rayleigh fading channel model. In *Proceedings IEEE ICUPC'94*, pages 160–164, 1994.

[WL83] Y. Wang and S. Lin. A modified selective repeat type II hybrid ARQ system and its performance analysis. *IEEE Transactions on Communications*, 31(5):593–608, May 83.

[WM95] Hong Shen Wang and Nader Moayeri. Finite-state Markov channel - a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, February 1995.

[WPD88] D. Waitzman, C. Patridge, and S. Deering. Distance vector multicast routing protocol. RFC 1075, November 1988.

[Wro97a] J. Wroclawski. Specification of the controlled-load network element service. RFC 2211, September 1997.

[Wro97b] J. Wroclawski. The use of RSVP with IETF integrated services. RFC 2210, September 1997.

[XP98] G. Xylomenos and G. Polyzos. IP multicast group management for point-to-point local distribution. *Computer Communications*, 21(18):1645–1654, 1998.

[YW95] James R. Yee and Edward J. Weldon. Evaluation of the performance of error-correcting codes on a Gilbert channel. *IEEE Transactions on Communications*, 43(8):2316–2323, August 1995.

[ZDE+93] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A new resource reservation protocol. *IEEE Network*, pages 8–18, 1993.

[ZRM95] M. Zorzi, R.R. Rao, and L.B. Milstein. On the accuracy of a first-order Markov model for data block transmission on fading channels. In *Proceedings IEEE ICUPC'95*, pages 211–215, November 1995.

[ZRM98] M. Zorzi, R.R. Rao, and L.B. Milstein. Error statistics in data transmission over fading channels. *IEEE Transactions on Communications*, 46:1468–77, November 1998.

Publications

Conferences and Workshops

- Neda Nikaein, Houda Labiod, and Christian Bonnet, ”MA-FEC: a QoS-Based Adaptive FEC for Multicast Communication in Wireless Networks”, Proceedings of IEEE ICC’00, New Orleans, USA, June 2000.
- Neda Nikaein, and Christian Bonnet, ”On the Performance of FEC for Multicast Communication on a Fading Channel”, Proceedings of IEEE ICT’00, Acapulco, Mexico, May 2000.
- Neda Nikaein, and Christian Bonnet, ”Le Multicast Sans Fil dans un Environnement IP”, Congrès DNAC (De Nouvelles Architectures pour les Communications), Paris, December 1999.
- George Fankhauser, Stathes Hadjiefthymiades, Neda Nikaein, and Lorraine Stacey, ” Interworking RSVP and Mobile IP Version 6 in Wireless Environments”, 4th ACTS Mobile Communication Summit 1999, Sorrento, Italy, June 1999.
- Neda Nikaein, and Christian Bonnet, ”Wireless Multicasting in an IP Environment”, Proceedings of IEEE MoMuc’98, Berlin, Germany, October 1998.
- Juha Ala-Laurila, Lorraine Stacey, Neda Nikaein, and Jukka Seppälä, ”Designing a Wireless Broadband IP System with QoS Guarantees”, 3rd ACTS Mobile Communication Summit 1998, Rhodos, Greece, June 1998.

IETF Draft

- George Fankhauser, Stathes Hadjiefthymiades, Neda Nikaein, and Lorraine Stacey, ” RSVP Support for Mobile IP Version 6 in Wireless Environments”, IETF Internet Draft, November 1998.

WAND Public Reports

- Lorraine Stacey, Juha Ala-Laurila, Jouni Mikkonen, Jukka Seppälä, Stathes Hadjiefthymiades, Neda Nikaein, George Fankhauser, and Sarantis Paskalis, ” IP over Wireless ATM”, ACTS Magic WAND Public Report, August 1998.
- Ioannis Dravopoulos et al., ” Wireless ATM MAC”, ACTS Magic WAND Public Report, August 1998.