

**DYNAMIC RESERVATION TDMA MEDIUM ACCESS
CONTROL PROTOCOL FOR WIRELESS ATM NETWORKS**

By

Jean-François Frigon

B.A.Sc. École Polytechnique de Montréal, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

August 1998

© Jean-François Frigon, 1998

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical and Computer Engineering

The University of British Columbia

2075 Wesbrook Place

Vancouver, Canada

V6T 1W5

Date:

August 31st

Abstract

To meet the anticipated demand for wireless access to the broadband Asynchronous Transfer Mode (ATM) network, the concept of wireless ATM has been proposed in 1994 [1]. One of the main challenge in the design of a wireless ATM network resides in the conception of a Medium Access Control (MAC) protocol that will handle the different ATM services while providing an efficient utilization of the wireless channel.

In this thesis, we propose a new Dynamic Reservation TDMA (DR-TDMA) MAC protocol for wireless ATM networks. DR-TDMA combines the advantage of distributed access and centralized control for transporting Constant Bit Rate (CBR), Variable Bit Rate (VBR) and Available Bit Rate (ABR) traffic efficiently over a wireless channel. The contention slots access is governed by the novel framed pseudo-Bayesian priority Aloha protocol that we introduce in this thesis. This protocol minimizes the contention delay and provides different access priorities for heterogeneous traffic. Simulation results indicate that the framed pseudo-Bayesian priority protocol offers a significant delay improvement for high priority packets with both Poisson and self-similar traffic, while low priority packets only experience a slight performance degradation. In the context of the DR-TDMA protocol, results show that the priority algorithm improves real-time traffic Quality-of-Service (QoS).

The DR-TDMA resource allocation algorithm grants to terminals reserved access to the wireless ATM channel by considering their requested bandwidth and QoS. We propose scheduling algorithms for CBR, voice, VBR and ABR traffic. We also introduce a method to dynamically adjust the number of uplink control slots per frame as a function of the estimated contention traffic. Furthermore, the DR-TDMA protocol features a

novel rate based allocation algorithm for VBR traffic and a cell control algorithm to determine the VBR flow conformance with the connection traffic parameters. Finally, an algorithm is proposed to integrate these algorithms to provide ubiquitous wireless ATM services. Performance results show that the DR-TDMA MAC protocol can achieve high throughput in the range of 90 to 95% while maintaining reasonable QoS for all ATM services.

Table of Contents

Abstract	ii
List of Tables	viii
List of Figures	ix
Acknowledgments	xiii
Chapter 1 Introduction	1
1.1 Motivations and Objectives	2
1.2 Structure of the Thesis	4
Chapter 2 Medium Access Control Protocol for Wireless ATM	5
2.1 Requirements and Constraints	6
2.2 Wireless ATM CDMA MAC Protocols	8
2.2.1 Multi-Code CDMA DQRUMA	8
2.2.2 Other CDMA Based MAC Protocols	11
2.3 Wireless ATM TDMA MAC Protocols	14
2.3.1 Broadband Access to a Wireless ATM LAN	15
2.3.2 Basic TDMA Reservation Protocols	17
2.3.3 Multiservice Dynamic Reservation TDMA	19
2.3.4 Other TDMA Based MAC Protocols	22
2.4 Comparison Between Wireless ATM MAC Protocols	23
2.5 Dynamic Reservation TDMA MAC Protocol	29

2.5.1	System Architecture Parameters	32
Chapter 3 A Pseudo-Bayesian Aloha Algorithm with Mixed Priorities		34
3.1	Background	35
3.2	Pseudo-Bayesian Slotted Aloha with Priorities	38
3.2.1	Pseudo-Bayesian Algorithm	39
3.2.2	Slotted Pseudo-Bayesian Priority Algorithm	40
	Idle Slot Case	42
	Success Slot Case	43
	Collision Slot Case	45
	Algorithm	48
3.3	Framed Pseudo-Bayesian Aloha with Priorities	50
3.3.1	Framed Pseudo-Bayesian Algorithm	50
	Algorithm	52
	Algorithm Evaluation	52
3.3.2	Framed Pseudo-Bayesian Priority Algorithm	57
3.4	Simulation Results with Poisson Traffic	58
3.4.1	Performance with Two Priority Classes	60
3.4.2	Performance with More than Two Priority Classes	69
3.5	Simulation Results with Self-Similar Traffic	72
Chapter 4 Integrated Resource Allocation Algorithm for DR-TDMA		79
4.1	Bandwidth Allocation for Wireless ATM	80
4.1.1	Scheduling Information	81
4.1.2	Review of Proposed Scheduling Algorithms	83
	CBR Allocation	84
	ABR Allocation	85

VBR Allocation	85
Integrated Traffic Allocation Algorithm	87
4.2 Multimedia Source Characteristics	88
4.2.1 Voice Source Model	89
4.2.2 Data Source Model	90
4.2.3 VBR Source Model	91
4.3 DR-TDMA Allocation Algorithms	96
4.3.1 Slot Allocation Algorithm for Voice Traffic	96
4.3.2 Slot Allocation Algorithm for Data Traffic	98
4.3.3 Slot Allocation Algorithm for VBR Traffic	100
4.3.4 Algorithm for Contention Period	105
4.3.5 Traffic Integration	108
4.3.6 Discussion on the DR-TDMA Allocation Algorithms	110
4.4 Simulation Results	113
4.4.1 Results for Voice Traffic Only	115
4.4.2 Results for VBR Traffic Only	119
4.4.3 Results for Data and Voice Integrated System	127
4.4.4 Results for Data, Voice and VBR Integrated System	136
 Chapter 5 Conclusion	 141
5.1 Summary	141
5.2 Topics for Future Investigations	145
 Bibliography	 147
 Appendix A Generation of Self-Similar Traffic	 154

List of Tables

2.1	DR-TDMA MAC frame parameters	33
3.1	Traffic class one waiting time in number of slots for slotted pseudo-Bayesian algorithms with self-similar traffic (90% confidence interval)	75
3.2	Average waiting time in number of slots for slotted pseudo-Bayesian algorithms with self-similar traffic (90% confidence interval)	76
3.3	Traffic class one waiting time in number of frames for framed pseudo-Bayesian algorithms with self-similar traffic (90% confidence interval)	77
3.4	Average waiting time in number of frames for framed pseudo-Bayesian algorithms with self-similar traffic (90% confidence interval)	77
4.1	Voice source model parameters	90
4.2	Frame parameters for simulations	114
4.3	VBR connections parameters	119
4.4	Parameters for data and voice integrated system simulations	128
4.5	Comparison between DR-TDMA and DRMA	136
4.6	Data connection parameters	137
4.7	VBR connection parameters	137

List of Figures

2.1	MC-CDMA with DQRUMA for multi-rate packet transmission protocol	9
2.2	Flow chart of the DQRUMA protocol at each mobile	10
2.3	CDM parallel transmission channel structure	12
2.4	Frame and sector structure	15
2.5	PRMA frame structure	17
2.6	MDR-TDMA frame structure	20
2.7	DR-TDMA wireless ATM MAC protocol frame structure	30
2.8	Uplink wireless ATM cell format	32
3.1	Wireless ATM frame structure	37
3.2	Waiting time in number of slots	54
3.3	Waiting time in number of frames	56
3.4	Average number of contention slots per frame	56
3.5	Waiting time as a function of the priority parameter for the SPBP system. $\lambda_1 = 0.18$ and $\lambda_2 = 0.18$ packets/slot.	61
3.6	Waiting time as a function of the priority parameter for the SPBP system. $\lambda_1 = 0.05$ and $\lambda_2 = 0.20$ packets/slot.	62
3.7	Waiting time as a function of the priority parameter for the FPBP system. $\lambda_1 = 0.15$ and $\lambda_2 = 0.20$ packets/slot.	62
3.8	Waiting time as a function of the priority parameter for the FAPBP system. $\lambda_1 = 0.05$ and $\lambda_2 = 0.30$ packets/slot.	63
3.9	Waiting time as a function of traffic class one arrival rate for the SPBP system. $\lambda_2 = 0.18$ packets/slot and $\gamma_1 = 1$	64

3.10	Waiting time as a function of traffic class one arrival rate for the FPBP system. $\lambda_2 = 0.25$ packets/slot and $\gamma_1 = 1$	64
3.11	Waiting time as a function of traffic class one arrival rate for the FPBP system. $\lambda_2 = 0.30$ packets/slot and $\gamma_1 = 1$	65
3.12	Waiting time as a function of traffic class one arrival rate for the FAPBP system. $\lambda_2 = 0.20$ packets/slot and $\gamma_1 = 1$	65
3.13	Cumulative density function of the waiting time for the SPBP system. $\lambda_1 = 0.15$ and $\lambda_2 = 0.20$ packets/slot. $\gamma_1 = 1$	67
3.14	Cumulative density function of the waiting time for the FPBP system. $\lambda_1 = 0.15$ and $\lambda_2 = 0.20$ packets/slot. $\gamma_1 = 1$	68
3.15	Cumulative density function of the waiting time for the FAPBP system. $\lambda_1 = 0.15$ and $\lambda_2 = 0.20$ packets/slot. $\gamma_1 = 1$	68
3.16	Waiting time as a function of the traffic class one priority parameter. $\lambda_1 = 0.08$, $\lambda_2 = 0.08$ and $\lambda_3 = 0.08$ packets/slot.	70
3.17	Waiting time as a function of the traffic class one priority parameter. $\lambda_1 = 0.05$, $\lambda_2 = 0.15$ and $\lambda_3 = 0.15$ packets/slot.	71
4.1	Bandwidth allocation problem	80
4.2	Cell Control Algorithm	94
4.3	Traffic Integration Flow Chart	109
4.4	Voice loss rate as a function of the number of voice connections ($R = 8.528$ Mbps).	116
4.5	Throughput as a function of the number of voice connections ($R = 8.528$ Mbps).	117
4.6	Voice loss rate as a function of the number of voice connections ($R = 664$ Kbps).	118

4.7	VBR cell loss rate as a function of the number of VBR connections (connections #1 and #2).	120
4.8	VBR cell delay as a function of the number of VBR connections (connections #1 and #2).	121
4.9	Throughput as a function of the number of VBR connections (connections #1 and #2).	121
4.10	VBR cell loss rate as a function of the number of VBR connections (connection #3).	122
4.11	VBR cell delay as a function of the number of VBR connections (connection #3).	122
4.12	Throughput as a function of the number of VBR connections (connection #3).	123
4.13	Comparison of the normalized guaranteed traffic load as a function of the number of VBR connections for different values of W .	126
4.14	Comparison of the cell loss rates as a function of the number of VBR connections for different values of W .	127
4.15	Voice loss rate as a function of the number of data connections with 100 voice connections ($R = 8.528$ Mbps).	129
4.16	Data waiting time as a function of the throughput with 100 voice connections ($R = 8.528$ Mbps).	129
4.17	Throughput as a function of the number of data connections with 100 voice connections ($R = 8.528$ Mbps).	130
4.18	Voice loss rate as a function of the number of data connections with 50 voice connections ($R = 8.528$ Mbps, mean batch length: 10 packets).	130
4.19	Data waiting time as a function of the throughput with 50 voice connections ($R = 8.528$ Mbps, mean batch length: 10 packets).	131

4.20	Throughput as a function of the number of data connections with 50 voice connections ($R = 8.528$ Mbps, mean batch length: 10 packets).	131
4.21	Voice loss rate as a function of the number of data connections with 25 voice connections ($R = 664$ Kbps, $CR = 1$, $CS_{max} = 1$).	132
4.22	Data waiting time as a function of the throughput with 25 voice connections ($R = 664$ Kbps, $CR = 1$, $CS_{max} = 1$).	132
4.23	Throughput as a function of the number of data connections with 25 voice connections ($R = 664$ Kbps, $CR = 1$, $CS_{max} = 1$).	133
4.24	Cell loss rate as a function of the number of VBR connections for the integrated system.	138
4.25	Throughput as a function of the number of VBR connections for the integrated system.	138
4.26	Delay as a function of the number of VBR connections for the integrated system.	139

Acknowledgments

I would like to express my appreciation to my supervisor, Dr. Victor Leung, for his guidance and suggestions throughout the course of this thesis. I also wish to express my gratitude to my colleagues, particularly Dr. Henry Chan and Hugo Jacques, for their help and comments throughout my studies and research. This work was supported by a postgraduate scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Finally, my thanks go to my family and friends, and specially Quyen Vu, for their encouragement and support throughout my graduate studies.

Chapter 1

Introduction

In the recent years, we have seen the development of two major trends in the telecommunication world: the evolution of the wireline network to support broadband multimedia services and the increasing success of personal communications systems. We can thus expect an increasing demand in the future to connect mobile devices to the broadband wired network.

Asynchronous Transfer Mode (ATM) has been proposed by CCITT to be the transfer protocol of the future *Broadband Integrated Services Digital Network* (B-ISDN) [2]. To extend the capabilities of ATM over the wireless channel, the concept or *wireless ATM* (WATM) was first proposed in 1994 [1]. In order to remain compatible with the wired ATM network, the wireless hop must support the standard ATM service classes: *Constant Bit Rate* (CBR), *Variable Bit Rate* (VBR) and *Available Bit Rate* (ABR) traffic.

However, the development of ATM assumed a fixed wireline network with characteristics that are not shared with the hostile wireless environment. For example, the first is characterized by a high bandwidth, low error rate and time-invariant channel while the second deals with a limited bandwidth, error prone and time-varying broadcast radio medium. Thus, the most crucial issues in the conception of an efficient WATM network are the design of the physical layer, data link layer, *Medium Access Control* (MAC) protocol and mobility management functions. In this project, we mainly focus on the design of an efficient MAC protocol and its associated resource allocation algorithm to satisfactorily handle the different ATM services and their *Quality of Service* (QoS) requirements

(bandwidth, access delay, cell delay variation, cell loss rate, ...).

1.1 Motivations and Objectives

In wired networks, adjacent nodes are joined by point-to-point communication links. Thus, the received message at one end of the link only depends on the transmitted message at the other end. On the other hand, in a *Wireless Local Area Network (WLAN)* the signal that is received at the base station consists of the sum of the transmitted message from a set of mobiles. There is thus a need in such an environment for a Medium Access Control protocol to efficiently and equitably allocate the multiaccess scarce radio medium among the competing mobile nodes. In a wireless ATM environment, the MAC protocol must support, at reasonable QoS levels, mobiles transmitting heterogeneous ATM traffic while maintaining a high radio channel utilization. Furthermore, the required bandwidth for many ATM services is time-varying; the wireless ATM MAC protocol must therefore adapt the channel allocation to these traffic variations.

In the past couple of years, several projects have developed MAC protocols for WLANs. Among, those we can highlight the IEEE 802.11 WLAN and HIPERLAN standard MAC protocols [3]. However, they can not be used as they are to provide a WATM service. Their main problem is the lack of priority for delay sensitive packets. Other MAC protocols have been specifically design to implement the WATM technology [4, 5, 6]. Three majors WATM prototypes were developed by Lucent [7], NEC [8] and the Magic Wand project [9]. Most of these protocols used centrally control demand assignment MAC protocols and are designed to support CBR, ABR and VBR traffic. However, few resource allocation algorithms that efficiently integrate these ATM traffic classes are proposed in the literature.

Further research must thus be conducted to develop a WATM MAC protocol and its

corresponding integrated scheduling algorithm that will *efficiently* support distributed CBR, ABR, voice and particularly VBR connections with their specified QoS. The main objectives of this thesis are therefore as follows:

- Design a MAC protocol structure that allows the multiplexing of distributed multimedia connections into a single radio channel within the area covered by a base station;
- Propose a contention algorithm with multiple priorities to reduce the access delay of time sensitive control packets and evaluate the impact of this algorithm on the performance of time sensitive connections;
- Develop a resource allocation algorithm that integrates CBR, ABR and VBR traffic;
- Evaluate the performance of the proposed protocol for different integrated traffic scenarios.

In order to meet these objectives, we propose a new **Dynamic Reservation Time Division Multiple Access** wireless ATM MAC protocol. The work conducted in this thesis differs from previous proposals in the following ways:

- The DR-TDMA protocol offers different contention access priorities according to the required QoS of the connection;
- The MAC protocol features a dynamic adjustment of the number of uplink control slots as a function of the estimated control traffic which allows a significant increase of the channel utilization for data transmission;
- We develop a novel rate based allocation algorithm to control the channel access of VBR station that can achieve a higher throughput than what has been previously reported;

- A cell control algorithm is proposed to determine cells that conforms to the VBR traffic parameter. Complying cells receive a guaranteed service while non-complying cells receive a best effort service.
- We propose a novel traffic integration algorithm that allocates the channel capacity to control and multimedia traffic based on the required QoS of the connections and their current traffic characteristics.

1.2 Structure of the Thesis

Chapter 2 investigates the proposed WATM MAC protocol in the literature. The characteristics of these protocols are analyzed in term of the constraints imposed by the wireless environment and the requirements of ATM traffic. The proposed *Dynamic Reservation TDMA* (DR-TDMA) MAC protocol is then introduced.

A novel pseudo-Bayesian priority Aloha algorithm is presented in Chapter 3. Theoretical analysis justifying the proposed algorithm are derived. An adaptation of the Aloha protocol to the framing environment is also described. Simulation results illustrating the performance of the protocol with Poisson and self-similar traffic are finally presented.

Chapter 4 proposes resource allocation algorithms for voice, ABR, VBR and control traffic as well as their integration for a complete CBR, ABR and VBR ATM traffic service that complies with the required QoS. The utilization of the framed pseudo-Bayesian priority Aloha (FPBP) algorithm described in Chapter 3 in the context of the DR-TDMA MAC protocol is also depicted. Performance results of the DR-TDMA protocol are given for different traffic scenarios.

Finally, Chapter 5 summarizes the majors points of the thesis and the results. Some suggestions for further avenues of research are also proposed.

Chapter 2

Medium Access Control Protocol for Wireless ATM

MAC protocols attempt to efficiently allocate the access to the radio channel among several competing distributed users. A key factor in the selection of a MAC protocol for wireless ATM is its ability to support CBR, ABR and VBR ATM traffic classes at reasonable QoS while maintaining a high utilization of the scarce radio medium. MAC protocols can be grouped into three classes [2]: fixed assignment, random access and demand assignment protocols. Fixed assignment techniques, like *Time Division Multiple Access* (TDMA) and *Frequency Division Multiple Access* (FDMA), involves a permanent channel assignment during the time of a connection. However, they can not take advantage of the bursty nature of ATM traffic and are thus inefficient for wireless ATM networks. Random access protocols like ALOHA can serve more efficiently bursty traffic, but the random nature of these protocols does not allow any control to ensure the QoS of the admitted connections. In demand assignment MAC protocols the channel capacity is assigned to users on a demand basis. The mobiles reserve bandwidth through a control channel and transmission capacity is allocated based on the transmission needs and required QoS of each admitted connections to achieve high multiplexing efficiency. This last group of MAC protocols are the most suitable to implement a wireless ATM network.

A MAC protocol also heavily depends on the nature of the multiple access technique used. FDMA is not considered for wireless ATM networks since it necessarily involves a fixed channel assignment. In a TDMA system, the access to the channel is divided on a time basis. That is, a user will occupy the whole channel bandwidth for a certain period of time and users are multiplexed by transmitting at different time. On the other hand, in a *Code Division Multiple Access* (CDMA) system a user sends information on the overall bandwidth during the whole message transmission time. Thus simultaneous transmission takes place at the same time. Multiplexing is achieved by spreading the transmission over the complete bandwidth using either a spread-spectrum sequence (i.e. the code) or frequency hopping.

In this chapter, we review several proposed WATM MAC protocol. We mainly focus on their MAC structure. Resource allocation algorithms are reviewed in Chapter 4. Requirements and constraints to which these protocols are subject are first highlighted. *Code Division Multiple Access* (CDMA) and TDMA based WATM MAC protocols are then described and compared, finally we introduce the DR-TDMA protocol that we have designed.

2.1 Requirements and Constraints

The design of an efficient WATM MAC protocol is subject to different requirements and constraints. An exhaustive list of requirements for wireless WLANs is presented in [10] and in [11] we can find some specific requirements to WATM MAC protocols. Among these we can underline the followings:

- High throughput;
- Low delay performance;

- Ability to serve data, voice and video with their required QoS;
- Preservation of packet order;
- Fair channel access algorithm.

Also, the wireless channel will impose some constraints on the MAC protocol design. Some issues related to WATM are presented in [7, 10, 12]. We can emphasize the followings:

- Equalizing or array antennae are needed to provide a high-speed radio channel due to multipath fading;
- Limited power at the mobile unit;
- Duplex mode;
- Difficulties to implement carrier sensing;
- Error recovery (retransmissions).

We should also stress the fact that usual wired LANs multiple access techniques such as *Carrier Sense Multiple Access* (CSMA) and *CSMA with collision detection* (CSMA/CD) are difficult to implement in the wireless environment [10]. The success of CSMA/CD in Ethernet relies on the ease of sensing the carrier by measuring the current or the voltage in the cable. However, the wireless environment does not permit collision detection by the mobile (it is impossible in a mobile to, simultaneously, send and receive information on the same radio channel). Furthermore, reliable carrier sensing required for CSMA techniques is extremely difficult due to severe channel fading in indoor environments, the use of directional antennas, the hidden terminal problem and co-channel interference. To overcome some of these problems, *CSMA with collision avoidance* (CSMA/CA) was

proposed. However, this protocol requires a four-way handshaking procedure and it presents a significant throughput reduction compared to the CSMA protocol.

2.2 Wireless ATM CDMA MAC Protocols

In a “pure” CDMA system, access to the channel is completely unconstrained so that a user begins transmission with its own spreading code whenever it receives new data. When the number of simultaneous transmissions increases, the multi-user interference increases causing a link quality degradation. It is obvious that no quality of service can be guaranteed with this kind of system. In fact, in order to guarantee any kind of QoS to the users, bandwidth allocation among the mobiles must be done centrally at the base station. This is achieved by a hybrid TDMA/CDMA system where simultaneous access by different users is provided on a slot by slot basis.

2.2.1 Multi-Code CDMA DQRUMA

A Lucent’s research group presents in [13] a multi-rate packet transmissions WATM CDMA/TDMA MAC protocol. It uses a *Multi-Code CDMA* (MC-CDMA) access technique while the *Distributed-Queuing Request Update Multiple Access* (DQRUMA) is the demand assignment medium access protocol. It has to be noted that a TDMA system based on the DQRUMA MAC protocol is also presented in [7, 14]. However, only the *Direct-Sequence CDMA* (DS-CDMA) based protocol will be analyzed (the TDMA protocol is really simple and is outperformed by many other protocols). This CDMA system is one of the most advanced and promising of all the WATM CDMA systems.

In a MC-CDMA system, data are transmitted over the radio channel only at the “basic” rate R_b . When a mobile needs to transmit at a rate \hat{m} times the basic rate, it converts its serial data streams into \hat{m} parallel basic-rate data streams. Each one is

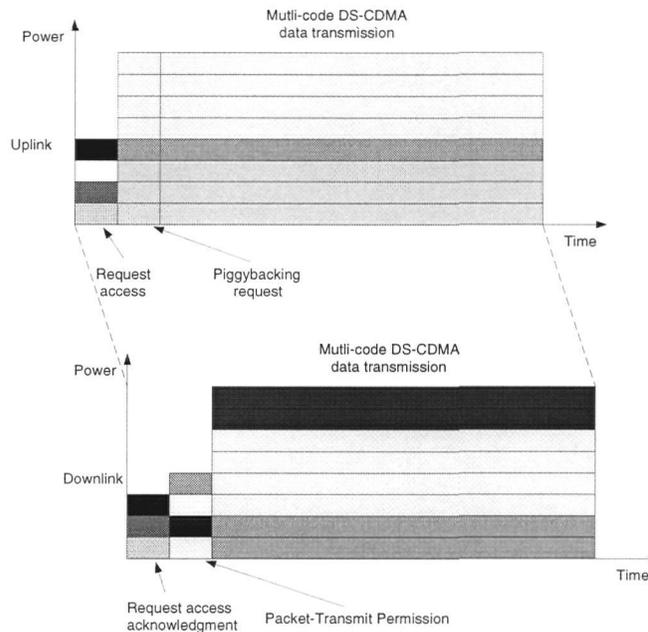


Figure 2.1: MC-CDMA with DQRUMA for multi-rate packet transmission protocol

then spreaded with a different code and they are superposed for radio transmission. The parallel spreading codes used by a mobile are generated by a sub-concatenation scheme that avoids self-interference. Thus, higher transmission rate sources will experience less interference than low-rate users. The transmitted power from high rate mobiles can therefore be lowered in order to increase the channel capacity.

Figure 2.1 shows the structure of a MC-CDMA time slot with DQRUMA for multi-rate packet transmission protocol. The uplink and downlink channels are assumed to be on different frequencies. In the downlink frame, the Request-Access Acknowledgment mini-slot acknowledges uplink request-access that occurred in the same time slot while packet-transmit permissions are for the next time slot. There is a time shift between the uplink and downlink frame due to transmission and processing delay.

Figure 2.2 illustrates the DQRUMA protocol flow chart. When packets arrive to a

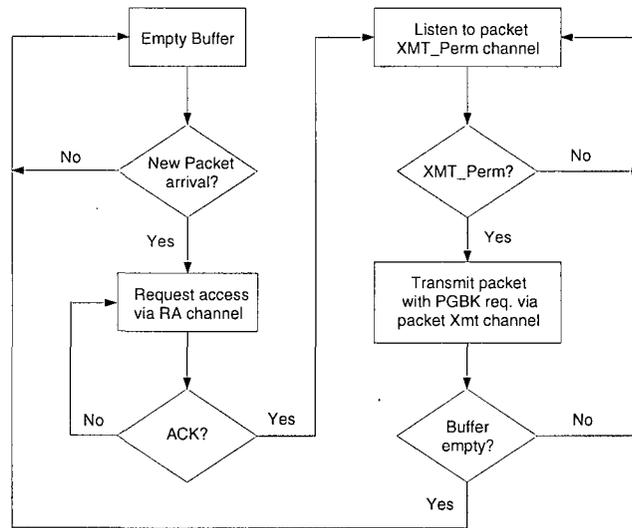


Figure 2.2: Flow chart of the DQRUMA protocol at each mobile

mobile with an empty buffer, the mobile sends a Request-Access packet in the uplink Request-Access mini-slot. The packet includes the mobile Access ID (assigned by the base station at the admission of the mobile in the system) and the number of packets for which the mobile is requesting to transmit. To transmit a Request-Access packet, a mobile randomly chooses one of the dedicated control PN codes. The number of dedicated PN codes can be chosen smaller than the number of mobiles admitted in the system which implies a form of random access in the Request-Access mini-slot.

When the base station successfully receives a Request-Access packet from a mobile, it sets the corresponding entry in the request table to indicate that the mobile has packets to transmit and acknowledges the reception on the downlink Request-Access Acknowledgment mini-slot using the mobile's spreading code. Once a mobile receives a positive acknowledgment that its Request-Access was received by the base station, it will listen with its despreading code at the downlink Packet-Transmit Permission mini-slots. A Transmit-Permission directed to a mobile includes a field indicating the number of

codes with which the mobile is given permission to transmit, a field assigning the primary PN code for the mobile's packet transmission in the next time slot and a field allocating the mobile's power level (for channel efficiency). Because the Transmit-Permission are announced on a slot-by-slot basis, primary codes can be assigned on a slot-by-slot basis to those mobiles given the Transmit-Permission. This assignment significantly reduces the number of PN codes required by the system. Transmit-Permission are allocated according to the desired packet transmission policy (not specified in the papers).

When a mobile receives a Transmit-Permission in a time slot, it configures its MC-CDMA transmitter according to the received parameters in the Transmit-Permission packet. In the next uplink time slot, the mobile transmits packets to the base station with the new transmitter configuration. The mobile can also send a contention-free Request-Access to the base station via the Piggybacking Request field in the data transmission packet.

2.2.2 Other CDMA Based MAC Protocols

Figure 2.3 illustrates the channel structure of the WATM MAC protocol presented in [15]. Similarly to the MC-CDMA protocol, the source bit stream is converted into parallel bit streams at the basic channel rate. Each basic bit stream is spreaded with a different spreading code before being superposed for wireless transmission. The physical channel is divided into logical channels. One of these is used as a signaling channel for the mobiles contending for access to the system. This channel is time slotted and the access is controlled by the slotted Aloha protocol. When the connection is established through the signaling channel, data transmission takes place on the other logical channels.

Let K denote the maximum number of basic parallel bit streams that a logical channel can support and r_b the basic rate of each parallel bit stream. Then each logical channel

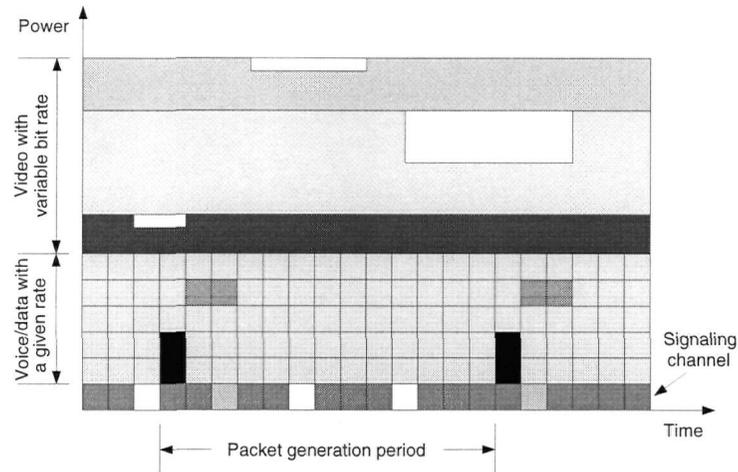


Figure 2.3: CDM parallel transmission channel structure

can provide a variable bit rate of kr_b ($k = 1, \dots, K$). Traffic channels are divided into two categories: multi-user channels for light traffic connections and single-user logical channels for heavy traffic connections. A multi-user logical channel has a TDMA structure. Periodic slots are assigned to constant bit rate connections like voice traffic while data traffic receives slots on an availability basis. For a single-user channel, the whole bandwidth (KR_b) is assigned to one mobile and it can transmit without taking care of time slot synchronization. In the case of CBR, a fixed number of data branches are used to keep a constant transmission rate. For VBR traffic, the number of data branches is variable in time from 0 to K (where K is the maximum transmission rate requested by the mobile). However, when the number of data branches used for transmission is less than K , the assigned bandwidth is wasted. The throughput is thus decreased but the interference on the other connections is reduced.

The MAC protocol proposed in [16] uses CDMA to divide the physical channel into different TDMA *Time Division Duplex* (TDD) channels. Each base station is assigned a group of channels for the transfer of ATM cells between the base station and the

mobiles within the base station cell. Delay-sensitive connections receive a unique control mini-slot that allows a contention-free transmission of high priority requests. On the other hand, request packets from low-priority connections are sent in contention using the Slotted Aloha protocol. After reception of the acknowledgment, the mobile waits for a transmission permission from the base station. When transmitting its data packets, the mobile can send contention-free requests to the base station piggybacked to the WATM cells.

A slotted, frame based protocol with CDMA transmissions in each time slot is presented in [17]. When a source wants to transmit information with a spreading code in a time slot (referred as a slot-code resource), it listens to the access field of the others while transmitting its own access field. If the slot-code is not used by an established non-queueable connection, the slot-code will be granted to the user with the highest priority access field. When several users of the same priority class have the right to obtain a slot-code, then there will be a collision and the slot-code will be wasted. While in this proposal the number of simultaneous transmissions in a time slot is limited, in the WATM MAC protocol introduced in [18], CDMA transmissions in a time slot are unconstrained. Therefore, no collision can occur, but when the number of simultaneous transmissions increases, the multi-user interference increases as well, and the probability that a packet is corrupted and lost is thus greater. Packet transmission from a user is controlled by a transmission probability (selected as a function of the packet priority) and the user state (i.e. idle, blocked or active). However, this system does not offer a good mechanism to control the connections' QoS.

2.3 Wireless ATM TDMA MAC Protocols

Most TDMA wireless ATM MAC protocols can be divided into two categories: polling MAC protocols and reservation based MAC protocols. Both approaches provide contention-free access to the TDMA channel on a time slot basis and the access is granted on a demand basis as the user needs it. In a polling system, the base station sequentially polls users for transmission privileges. In some systems, users are polled both for request and data transmissions, while in other systems requests are sent using a random access protocol. The poll signal can be used to set parameters of anti-fading devices (antennas weight, equalizer coefficients, ...) which can help to increase the link quality and therefore the channel capacity. However, the efficiency is reduced by the time wasted for poll signal transmissions. Furthermore, the mobile must listen over the channel most of the time to hear its poll. Little idle time is thus available for power saving or channel scanning.

In reservation TDMA MAC protocols, as in polling systems, users send request packets (either with a random access or a contention-free protocol) to reserve non-contending slots for actual data transmission. However, in reservation protocols the time slots are grouped into frames and the base station makes an announcement of the frame's slot allocation such that mobiles know in advance when they will transmit and receive their information. In some systems, the reservation is implicit, that is if a user successfully transmits in a time slot, in subsequent frames the same time slots will be reserved to this user until the end of the message transmission. On the other hand, the slot reservation announcement can be explicit when at the beginning of each frame the base station explicitly assigns each slot to a mobile. Reservation protocols have a better throughput (less overhead) and mobiles can perform other operations while waiting for their transmission or reception time slots. However, since slot allocations are done in advance, it allows

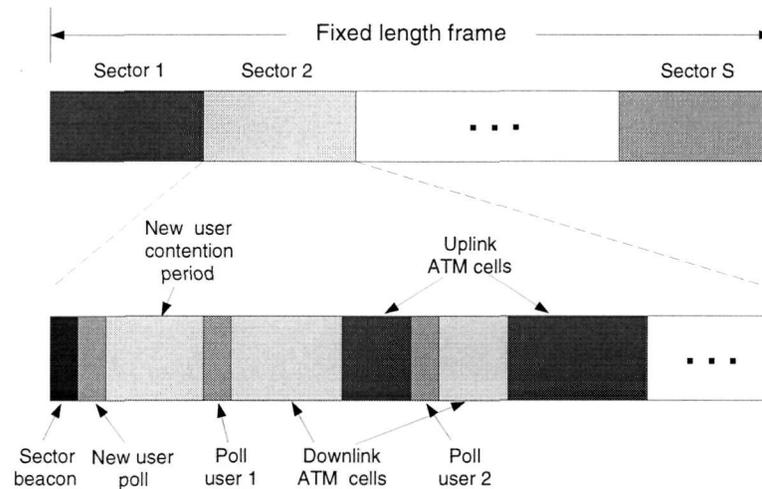


Figure 2.4: Frame and sector structure

less flexibility in packet scheduling compared to polling protocols where instantaneous decisions can be taken (i.e., retransmission of an important packet, new high priority request, ...).

2.3.1 Broadband Access to a Wireless ATM LAN

The WATM polling MAC protocol proposed in [19, 20] is predicated on the use of switched, multi-sector antennas at the base stations and mobiles. These antennas, which permit transmission and reception on only one antenna beam at a time, serve to focus power and thus reduce transmitter power requirements, multipath fading and inter-cells interference.

Figure 2.4 shows the overall fixed length frame structure of the protocol. The frame is divided into sector segments (not necessarily of the same lengths), one for each of the base antenna sectors. During each sector segment, the base station communicates with mobiles registered in that antenna sector. The order of the sector segments may vary

from frame to frame to ensure that co-channel interference is independent from frame to frame.

A sector segment begins with a beacon envelope that is used by mobiles to determine the best combination of their own and base antenna sector for transmission with their base station. When a mobile is new and seeking to register with a base station or if it is already registered on a base/sector combination, but is evaluating possibilities for a future hand-off to a new base station or sector with a better link quality, it will passively monitor each of its own mobile antenna sectors listening for base antenna sectors beacon envelop.

Following the beacon envelop are registration slots used by mobiles wishing to register to this base/sector combination to transmit request packets with the slotted Aloha protocol. A request packet contains information about the rate the mobile wishes to be polled, its traffic class priority and the transmit power level it requests from the base station. Acknowledgments to the requests will be sent in the new user polling period of the next frame.

If the base station has any packets intended to all the mobiles, a control envelop will then be sent to poll all the mobiles and the broadcast packets will be transmitted. Then, the base station starts polling individual mobiles. Whether a mobile is polled or not depends on its current assigned polling rate, on the time since its last poll and its traffic class priority. Each polling packet contains the maximum number of uplink packets the base will accept, the number of downlink packets to be sent by the base station and acknowledgments for the previous frame uplink packets from the mobile. After the poll, the base station sends its packets and the mobile answers with a control packet followed, if it has something to send, by its uplink packets up to the maximum allowed. Included with the control packet will be acknowledgments for downlink packets, request for a polling rate and/or power level change and request for a hand-off to a different base



Figure 2.5: PRMA frame structure

antenna sector.

This scheme is very efficient to reduce the co-channel interference. We can thus use a smaller frequency reuse factor and increase the system capacity. However, there is a lot of overhead and the throughput is therefore reduced. Furthermore, no indication on the slot allocation algorithm for this scheme is reported.

2.3.2 Basic TDMA Reservation Protocols

Most of the wireless ATM reservation protocols are derived from the basic reservation protocols designed for voice and data traffic that we describe in this section. The *Packet Reservation Multiple Access* (PRMA) MAC protocol [21] is one of the most popular voice/data reservation protocols. Figure 2.5 illustrates the PRMA frame structure. Slots in a frame are either available or reserved by a voice user. Voice users that have a newly generated talkspurt but have not yet obtained a reservation and active data users can contend for the available slots according to the voice and data transmission probabilities (chosen to give a higher priority to voice users). After an available slot, the base station broadcasts the outcome of the contention (collision, idle or success by which type of user). If a voice user succeeds, this slot will be labeled as reserved and the voice user can transmit in the same slot in subsequent frames until the end of its current talkspurt. If a data user succeeds in the contention, this slot is still marked as available and the data

user can only use the slot in the current frame and must contend again to transmit further data packets. In PRMA, no dedicated reservation bandwidth is used. Therefore, if all the slots are reserved no request can be made. PRMA also uses an information packet to make a channel reservation. This is somewhat inefficient in that each unsuccessful reservation wastes a whole time slot.

Some modifications to PRMA have been proposed to overcome some of its shortcomings. In the *PRMA-Independent Stations Algorithm* (PRMA-ISA) protocol [22] an algorithm to manage the access rights in available slots that maximizes the success probability is presented. The Multi-Rate PRMA protocol [23] supports sources with different bit-rates as opposed to the PRMA protocol where the maximum transmission rate for a voice user is one slot per frame. In the *Dynamic Reservation Multiple Access* (DRMA) protocol [4], available slots are divided into a set of mini-slots. If a user chooses to transmit in the available slot, it will randomly select one mini-slot to send its reservation packet. Each successful user will be assigned one of the available information slots in the frame (including the current slot starting from next frame). Short reservation packets in DRMA can utilize the system bandwidth more efficiently than information packets in PRMA. A scheme similar to DRMA is used in Centralized-PRMA [24], however the protocol uses the request information to poll users according to their needs.

In Dynamic-TDMA [4] and in PRMA++ [25], the uplink period is divided into reservation and traffic slots. A user transmits a short request packet in the reservation slots using the slotted Aloha protocol. Bandwidth is always dedicated in each frame for making reservations and it has been suggested to dynamically adjust length of the reservation period to achieve the best performance [4]. Acknowledgments from the base station indicate mobiles that were successful in the contention and those that are assigned a slot in subsequent frames until the end of the talkspurt. Mobiles that did not receive a slot allocation but successfully transmitted the request to the base station will continue to

monitor acknowledgment slots until they receive a slot reservation. This has multiple advantages: smaller slots can be used for reservation, access to the system is always possible even under heavy load and the base station has centralized control over the traffic slot allocation policy (this is particularly important when different priority services are contending). The *Resource Auction Multiple Access* (RAMA) protocol [26] uses an auction procedure to access the reservation mini-slots. The auction procedure is based on the user's ID (can be chosen to allow different priorities). During the auction slot, requesting users will transmit their ID's one bit at a time. Following each transmission, the base station announces the highest value received on the channel and only users which sent this values will continue the auction procedure. At the end of each auction slot, there will be a single winner who will not attend future auction. At the end of the auction slots period, the base station announces the final resource assignment. With this scheme the reservation success probability is much higher. However, it is not obvious how the auction procedure can be implemented in a high speed channel.

2.3.3 Multiservice Dynamic Reservation TDMA

Many WATM MAC protocols based on TDMA reservation that have been proposed in the literature are similar to the *Multiservice Dynamic Reservation TDMA* (MDR-TDMA) protocol proposed by NEC [1, 27]. This protocol will be described and then some variants are introduced.

Figure 2.6 outlines the MDR-TDMA frame format. MDR-TDMA dynamically divides the uplink and downlink channels using *Time Division Duplex* (TDD) as a function of the traffic load. Some other authors prefer to separate the two channels using *Frequency Division Duplex* (FDD). The modem preamble is used to perform diverse radio physical layer functions while the frame header announces the boundaries between the frame

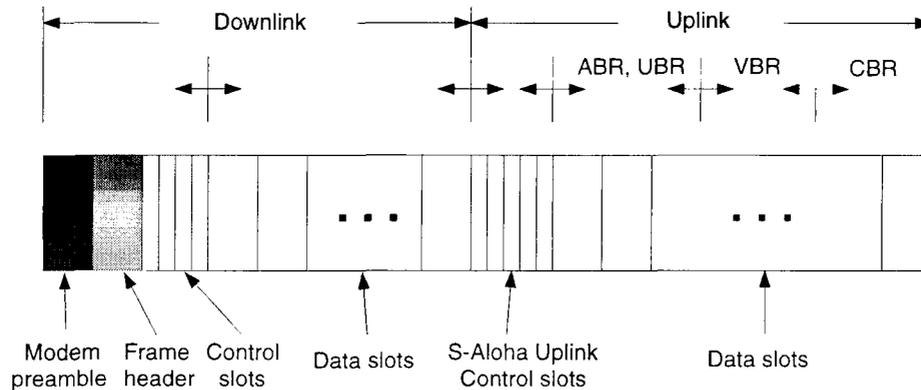


Figure 2.6: MDR-TDMA frame structure

periods. The base station has absolute control in determining the number of slots in each part of the frame and which mobile will receive or send information during the data slots. This is done according to the chosen scheduling algorithm.

In the uplink period, control slots provide a communication mechanism for a mobile station to send a reservation request, while the data slots supply it with resource bandwidth during the network connection. Uplink control packets are sent in a slotted Aloha contention mode by transmitting the packet in a randomly selected control slot. Acknowledgments for the uplink control slots are transmitted in the next downlink control packets. Downlink control packets contain as well announcements for downlink data slot transmissions and allocations of uplink data slots to mobiles. If a collision occurs, the mobile will again contend in the next frame. After completing the contention procedure, the mobile station can use the data slots without undergoing further contentions while it has data to transmit.

Uplink control packets provide the base station with the traffic characteristics of the connection. Updated information can also be sent piggybacked to the data packets. The base station will use these traffic parameters to allocate data slots to each reserving

station. When a connection is successful in the contention period, the mobile monitors the control slots in each subsequent frame to receive its frame slot assignment. After receiving this information, the mobile transmits the WATM cells in its assigned slots in the frame.

CBR connections are assigned slots periodically according to their bit rates specified when the new (or hand-off) calls are established. Furthermore, the position of the assigned slots within a frame are maintained relatively static to facilitate operation of low complexity terminals and to reduce the downlink control signaling. ABR slots are assigned dynamically on a frame-by-frame basis to provide a best-effort service [28]. VBR service may be provided with a suitable combination of these periodic and dynamic allocation modes or by using an adaptive slot allocation scheme [29].

Some variants of the MDR-TDMA protocol have also been proposed for WATM MAC protocol [5, 30, 31, 32, 33, 34]. We will now highlight some of the differences between these protocols and the MDR-TDMA protocol. In *PRMA with Dynamic Allocation* (PRMA/DA) [30], an algorithm is proposed to dynamically adjust the number of uplink control slots depending on the congestion state of the network. The objective of the scheme is to maximize the throughput by having the majority of slots serving the reserving stations while maintaining a minimal number of request slots to allow network access by contending stations.

In *Dynamic Slot Assignment* (DSA++) [31, 35, 36], instead of using slotted Aloha as the access protocol for uplink control slots, they proposed the utilization of a tertiary splitting algorithm with a higher throughput. Some proposals are also made to provide a faster feedback to reduce the system access delay.

A contention-free scheme, using time orthogonal codes, to send uplink control request packets is proposed in *Dynamic Slot Allocation Multiple Access* (DSAMA) [33]. However, the request slot length determines the maximum number of simultaneous users in the

system. Furthermore, the request orthogonal code can not be used to transmit traffic parameters to the base station.

In the *Dynamic Slot Multiple Access* (DSMA) protocol [5] when a connection is established a unique status slot is assigned to the connection if it needs to transmit control information (i.e. VBR and ABR connections). This guarantees a contention-free transmissions of request packets. Similarly, for the *Non-collision PRMA* (NC-PRMA) wireless ATM MAC protocol [34], the uplink slots are divided into assigned control slots for contention-free access of time-sensitive control packets and random access slots for non time sensitive control packets. However, for both protocols, the number of admitted connections is limited by the number of contention-free control slots available in the frame.

2.3.4 Other TDMA Based MAC Protocols

Acompora introduces a WATM polling MAC protocol in [12]. A mobile having queued ATM cells sends a request message in its assigned polling slot. The base station immediately acknowledges this request in the base-to-mobile subsegment of that polling slot. Receipt of the mobile's request at the base station reserves for this mobile the corresponding time slot in the uplink period. If the mobile has not sent a request, the base station may assign the time slot for other purposes (other users, control, ...). At the beginning of a reserved uplink time slot, the mobile sends a brief pilot tone of RF (used for adjustment at the base station) followed by an ATM cell. In the downlink period, the base station polls intended mobiles, that replies with a pilot tone of RF. The base station then immediately sends its ATM cells sequence to the mobile. This MAC protocol is very efficient to deal with the time-varying channels. However, this protocol can not handle integrated traffic and different bit-rate connections.

Another TDMA based polling system is presented in [37]. A polling signal preceding each time slot announces if the slot is a downlink or uplink slot and, for the latter case, which mobile is authorized to transmit or if it is a reservation slot. The reservation slots are divided into mini-slots, the first half is accessed using a slotted Aloha protocol and the second half is used by the base station to send acknowledgments.

Two others WATM MAC polling systems are presented in [11]. During the request phase, the base station polls all inactive terminals to allow further transmissions without undergoing a contention phase. These schemes are different from the Acompora scheme where all the stations (active and inactive) transmit during the request slots. In the proposed protocols, active stations piggyback information about their bandwidth requirements for the next frame in the information packets that are sent to the base station.

A token MAC protocol approach is proposed for the SWAN system [38, 39]. However, the system is design such that only one mobile has access to a given physical channel during the whole length of its connection and the token is just intended to transfer the transmission privilege from the base station to the mobile user. Finally, with some modifications, the *Group Randomly Addressed Polling* (GRAP) protocol [10] can also be used as a WATM MAC protocol.

2.4 Comparison Between Wireless ATM MAC Protocols

For most of the proposed systems in the literature, some performance evaluations are presented. However, it is really hard to compare these systems based on their performance results since different assumptions were taken (channel bit rate, source characteristics,

overhead, ...). Many papers do not make any comparisons with other systems to evaluate the performance of their proposal and, if a benchmark system is used, it is often the PRMA protocol (which is not too hard to outperform). Although, we can compare these protocols based on their general properties.

All proposed wireless ATM MAC protocol can be classified by the multiple access and duplex method that they utilize. For multiple access methods, the following systems can be considered:

- Time Division Multiple Access system:
 - Polling protocol;
 - Reservation protocol.
- Code Division Multiple Access system:
 - “Pure” CDMA protocol;
 - Hybrid reservation TDMA/CDMA protocol.

For the duplex method between uplink and downlink transmission, there are two options:

- Frequency Division Duplex (FDD);
- Time Division Duplex (TDD).

In a FDD system, the total bandwidth allocation for uplink and downlink connections is determined by the frequency band attributed to both components. It is therefore easy to manage co-channel interference in a FDD system. However, this frequency allocation is static while bandwidth requirements for uplink and downlink ATM connections are dynamic. It is thus clear that FDD is not suitable for a WATM network. On the other hand, a TDD system can dynamically adapt to the instantaneous connections

requirements. Furthermore, since we have a short propagation time, guard time needed for a TDD system is short. TDD also offers the advantage that uplink and downlink channels have the same radio propagation characteristics (fading, delay, ...), which is not the case with FDD. Thus, for example, anti-fading information (equalizer coefficients, directional antenna weights, ...) gathered on the uplink can be used for the following downlink transmission. Therefore, TDD is preferable for a WATM network.

As explained in Section 2.2, a “pure” CDMA system can not be used for wireless ATM networks with QoS requirements. In polling systems, a polling signal from the base station always precedes a mobile transmission. However, the polling signal increases the overhead and the throughput is therefore reduced. Furthermore, in a high speed TDD channel, the guard time required and the time to switch from transmission to reception in the base station will result in important bandwidth loss. Polling systems are therefore not appropriate for WATM networks. For the same reasons, the base station can not send a feedback for an uplink transmissions immediately after each time slot. Therefore, any protocol relying on an immediate channel feedback will be inefficient in the wireless ATM environment.

There is thus only two systems that are appropriate for the WATM environment: the hybrid reservation TDMA/CDMA system and the reservation TDMA system. There is no general agreement on whether TDMA or CDMA is the most efficient multiple access method for the voice-only digital cellular radio and even less for wireless ATM networks. In [40] and [41] a comparative study of a “pure” packet CDMA system versus a *Dynamic-TDMA* (D-TDMA) network in an integrated voice/data environment is presented. Although these protocols are not suitable for a WATM environment, it is worthwhile to look at the main conclusions of the study.

From a system point of view, it is observed that CDMA random access mode provides

a zero channel access delay and an almost perfect statistical multiplexing of traffic. However, the simultaneous CDMA transmissions appear as interference at the receiver and a power control system must be used to avoid near-far interference. Under increasing load there will be a quality degradation, this is in contrast to D-TDMA where the quality is constant but congestion leads to higher probability of voice blocking and longer delay for data messages. Finally, the comparison between CDMA and D-TDMA involves a number of factors that are inherent to the multiple access method used. We can underline the followings:

- Spatial re-use in TDMA requires explicit frequency coordination while in CDMA it is achieved implicitly. This also implies that “soft hand-off” between cells is possible with CDMA while in TDMA complex “hard hand-off” procedures are needed;
- CDMA provides a signal quality advantages in the presence of interference and multipath transmissions while with TDMA a fairly complex equalizer is required;
- The burst data rate on a TDMA channel can be higher than in a CDMA system. So message transmission delay is generally longer in a CDMA system;
- Timing system is more complex in a TDMA system than in a CDMA system where user synchronization is minimal;
- Higher data rates in TDMA are associated with larger peak transmit power requirements at the remote terminal than for CDMA;

Simulation results show that the packet CDMA protocol does have a higher capacity than the D-TDMA protocol. However, the CDMA system performance advantage is very sensitive to the assumed propagation loss constant. It is also observed that the CDMA protocol provides good delay performance for short data messages but higher

data delay than D-TDMA for longer data messages generally associated with file transfer and multimedia applications.

For WATM networks, we must compare the hybrid reservation TDMA/CDMA system and the reservation TDMA system. Since a slotted structure is used for TDMA/CDMA systems, some advantages associated to the CDMA multiple access method are lost (synchronization, complexity, ...) while others are preserved (lower mobile power, frequency management, ...). We will compare two systems that are representative of both types of WATM MAC protocol: the MC-CDMA system (Section 2.2.1) and the MDR-TDMA system (Section 2.3.3). We assume that for both system, the amount of information transmitted with M spreading codes during a period of time T is the same as the information transmitted in M slots of length T/M .

First, we can observe that the downlink period for each system is composed of a downlink control sub-period (acknowledgments for uplink control and information packets, uplink transmission permissions and downlink transmission announcements) and a data transmission sub-period. Similarly, in the uplink period, we find in each protocol a request transmission period and a data transmission part.

In MC-CDMA, when a mobile wants to transmit an uplink control packet it chooses between M control spreading codes and if more than two users choose the same code, a collision occurs. While in the MDR-TDMA system, a mobile chooses between M control slots and a collision occurs when more than two users transmit in the same control slot. Downlink control packets, in the MC-CDMA system, are transmitted using the N downlink control codes as the spreading sequence, while in MDR-TDMA, transmissions are done in the N downlink control slots. Finally, uplink (downlink) data are transmitted using the spreading code allocated (announced) by the base station in the downlink control information for the MC-CDMA system, while in MDR-TDMA, data packets are transmitted in the time slots allocated (announced) by the base station.

Therefore, if we replace spreading codes by time slots, the two systems are almost identical. Yet, some differences remain. First, the MC-CDMA system uses FDD to separate the uplink and downlink channel, but there is nothing that prevents us to modify it to obtain a TDD system as for the MDR-TDMA system. Second, the TDMA/CDMA system is less flexible for bandwidth allocation since the number of codes available in a time slots is a fixed system parameter. Therefore, this creates a higher granularity for bandwidth allocation among control, data, uplink and downlink slots. Furthermore, it is believed that CDMA systems are more complex to implement in a high speed environment and offer lower maximum bit-rate limit [1].

On the other hand, in the MC-CDMA system, when not all the codes are used, the link quality of transmitting users is increased, thus reducing the probability of reception errors and retransmission (hence reducing the traffic on the channel). Furthermore, it offers the possibility to add some codes when there are urgent packets (retransmissions, expiring time-sensitive packets, ...) to transmit at the expense of a decreasing link quality. However, we believe that this increase in the number of spreading codes should not be accepted in a wireless ATM protocol because the quality of service of transmitting users can not be guaranteed.

We believe that the advantages of a reservation TDMA/TDD structure (flexibility, higher bit-rate, ...) over a hybrid TDMA/CDMA system make it a better choice to implement our WATM MAC protocol. Furthermore, most of the proposed WATM MAC protocols use a TDMA based reservation MAC protocol. It is thus preferable to continue the research in this direction. However, as explained before, if we replace the time slots by spreading codes, the MAC protocol that we propose can be adapted to a CDMA structure.

2.5 Dynamic Reservation TDMA MAC Protocol

A centrally controlled dynamic reservation TDMA/TDD approach is adopted for our MAC protocol to multiplex multimedia connections into a single radio channel within the area covered by a base station. The DR-TDMA protocol that we introduce is based on the MDR-TDMA MAC protocol [1]. To improve the MDR-TDMA system performance, we have focused on the following points in the design of the DR-TDMA MAC protocol:

- Provide a mechanism to manage the control slots access such that contention delay can be minimized and priority services can be provided without limiting the number of users;
- Dynamically adjust the number of uplink control slots per frame to optimize the channel utilization;
- Design an efficient resource allocation algorithm to integrate CBR, VBR and ABR traffic.

Figure 2.7 illustrates the frame structure of the DR-TDMA MAC protocol for supporting multi-service traffic. The DR-TDMA MAC frame is time duplex into an uplink and downlink channel and the boundary between these two periods is dynamically adjusted as a function of the traffic load. However, the total frame length is kept fixed in order to facilitate coordination between cells, hand-offs and the operation of low complexity terminals. Downlink and uplink channels are further divided into control and data transmission periods (the control and data periods designate, respectively, the slots assigned for control and data transmissions in the downlink or the uplink channel). The uplink and downlink control period lengths (the length refers to the number of slots assigned for this control period in the frame) are also dynamically adjusted as functions of the control and data traffic loads. The base station has absolute control in determining

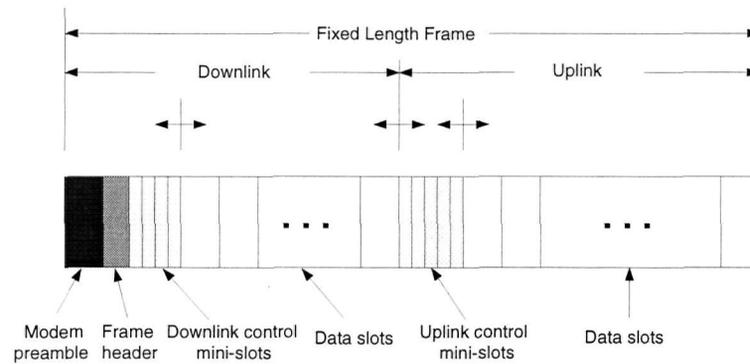


Figure 2.7: DR-TDMA wireless ATM MAC protocol frame structure

the number of slots in each part of the frame and which mobile will receive or send information during the data slots. This is done according to the resource allocation algorithm presented in Chapter 4.

The modem preamble is used by radio physical layer functions to perform diverse operations: synchronization, power control, hand-off monitoring, etc. The frame header announces the frame boundaries between downlink control slots, downlink data slots, uplink control slots and uplink data slots. A control slot designates a slot assigned for control purpose. The control slots are divided into control mini-slots used to transmit control packets. A data slot defines a slot used for data transmission (either CBR, VBR or ABR).

Downlink control packets contain acknowledgments for uplink control and data cells, announcements for downlink data slot transmissions and allocations of uplink data slots to mobiles. Other network management functions can also be done in this downlink control period like announcement of uplink control packets transmission probabilities, power management, answer to hand-off requests, ...

In the uplink channel, control slots provide a communication mechanism for a mobile station to send a reservation request, while the data slots supply it with contention free

resource bandwidth during the network connection. An uplink control packet is sent when a mobile: wants to establish a new connection, has a new data message to send, wants to increase the bandwidth allocated by the base station, begins a new talkspurt, . . . Uplink control packets are sent in contention according to the framed pseudo-Bayesian priority Aloha algorithm introduced in Chapter 3. With this protocol, contention transmission priorities are assigned to mobile stations according to the required QoS of their connections. The contention delay of time sensitive control packets is thus reduced. Feedback for the uplink control slot transmissions are sent in the downlink control packets in the next frame. Mobiles for which control packets experienced a collision will again contend in the next frame. After completing the contention procedure, the mobile station can use the data slots assigned by the base station without undergoing further contentions (unless one of the situations mentioned above arises).

When there is no uplink traffic from a mobile but downlink packets are sent to it, uplink control packets can also be used to acknowledge downlink packets and to adjust anti-fading parameters at the base station for this mobile. These uplink control mini-slots are directly assigned to mobiles by the base station and are therefore accessed without any contention.

Uplink control packets provide the base station with the traffic characteristics and source status of the connection. Status information can also be sent piggybacked to the data packets. The base station will use these traffic parameters to allocate uplink data slots to each reserving station. When a connection has successfully sent its request, it monitors downlink control slots in each subsequent frames to receive its frame slot assignment. If the mobile receives a slot assignation, it transmits packets in the allocated uplink slots in the current frame.

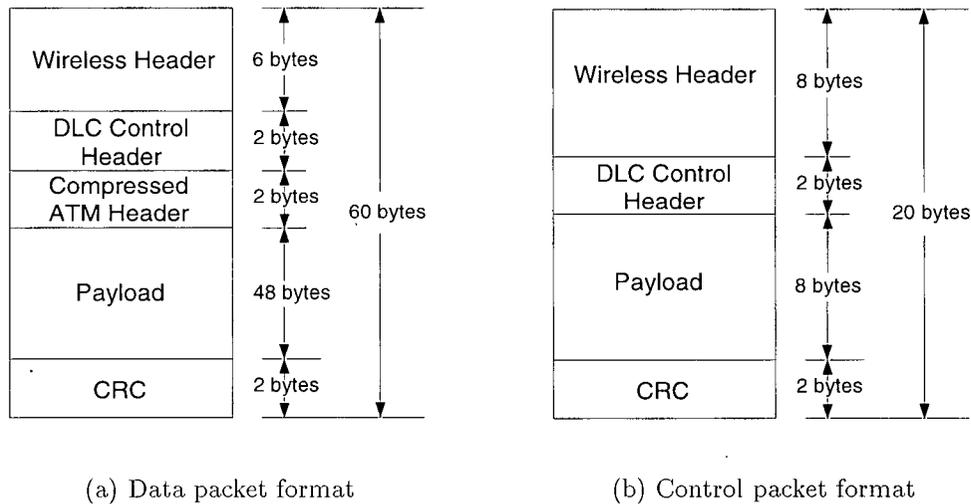


Figure 2.8: Uplink wireless ATM cell format

2.5.1 System Architecture Parameters

System architecture parameters adopted for the DR-TDMA protocol are according to the literature [1, 29]. Figure 2.8(a) illustrates the format of uplink wireless ATM data packets. The overhead is composed of a 2 byte DLC control header (sequence number, service type, hand-off indicator), 2 byte compressed ATM header, and 2 byte FEC/CRC used to correct and/or detect errors depending on the payload. The 6 byte wireless header includes bits for synchronization, guard time, anti-fading information (i.e. equalization), in-band signaling for piggybacking status information, and forward error correction. Figure 2.8(b) shows the format of an uplink control packet. The wireless header is longer since uplink control packets are usually from users that have not transmitted cells to the base station for a long time. Better anti-fading and error correction methods are thus required. Each slots in the frame has the same length as data packets (i.e. 60 bytes). When a slot is assigned for control purpose, it is divided into 3 control mini-slots.

Table 2.1: DR-TDMA MAC frame parameters

Channel bit rate	8.528 Mbps
Frame duration	2 ms
Uplink data slot size	60 bytes
Uplink control slot size	20 bytes
Preamble size	16 bytes
Frame header	16 bytes
Number of slots per frame	35
Number of control mini-slots per slot	3

Downlink packets (both data and control) are similar to uplink packets; however the wireless header can be reduced since the modem preamble at the beginning of the frame plays the same role [8]. Slot allocations and cell acknowledgments for all mobiles are grouped in the same downlink control slots (i.e. each downlink control slot is not dedicated to a specific mobile). However, the exact structure of downlink packets is not addressed here since in this work we are more concerned about the performance of the DR-TDMA protocol for the uplink period of the frame as explained in Section 4.3.

Table 2.1 summarizes the DR-TDMA MAC frame parameters. It should be noted that the only parameters that directly influence the MAC protocol efficiency are: the frame duration, the number of slots per frame and the data to control slot size ratio.

Chapter 3

A Pseudo-Bayesian Aloha Algorithm with Mixed Priorities

In the DR-TDMA MAC protocol, as in all reservation protocols, a mobile sends a request to the base station to obtain a contention-free access to the wireless channel. However, the request packet is transmitted in contention with request packets from other connections. Thus, one of the most critical phase for quality of service and delay in a reservation MAC protocol is the contention period that a connection goes through before obtaining a contention-free channel access.

For certain classes of traffic the contention phase is the limiting part of the scheme [21, 29] while other classes are not too sensitive to the introduced delay. The first packet of a voice talkspurt, a request for new bit-rate in a VBR connection or a hand-off request for a real-time connection are examples of control packets that are very sensitive to contention delay. Long contention time can result in loss of packets, buffer overflow, dropping of connections, ... On the other hand, control packets for data messages or requests to establish new connections are less sensitive to the contention delay. It is thus necessary to find a contention access protocol that will give a higher priority to delay sensitive packets in order to implement an efficient multimedia wireless ATM MAC protocol.

In the next section, we will review some random access protocols with mixed priorities

proposed in the literature. In Section 3.2, the slotted Aloha protocol with mixed priorities is presented. A modification to adapt the protocol to the framing environment is then introduced in Section 3.3. Simulation results of the protocol with Poisson and self-similar traffic are presented in Section 3.4 and 3.5. Finally, it has to be noted that the priority protocols proposed in this chapter are not just specific to WATM but can be used in any situation where there is multiple traffic streams requiring their own quality of service that are contending for the same channel.

3.1 Background

Several multiple access protocols for a system with mixed priorities are presented in the literature. A first group of priority protocols avoids collisions between high and low priority packets [42, 43, 44, 45]. In [42] and [43] the low priority traffic is embedded into the primary high priority traffic contention protocol. In [42] a slotted Aloha protocol is used and the low priority traffic find the idle slots in the high priority traffic transmissions via CSMA. In [43], a well-defined finite-number user population is assumed and time slots are grouped into fixed length frames. A deterministic tree-search is performed for high priority traffic contention. Through channel feedback, low priority users detect the end of the contention procedure for high priority traffic. Low priority traffic then contends for the slots left in the frame using the random access algorithm proposed in [46].

In [44] and [45, 47], low-priority transmissions are postponed to avoid collision with high priority packets. When the protocol detects that there is no high priority packets in the system (i.e. no high-priority transmissions), low-priority users are allowed to transmit. Furthermore, a transmission permission window is used to give a service advantage to high-priority traffic. The window length is smaller for low priority traffic in order to reduce the contention resolution period such that high-priority traffic contention can

restart earlier. In [44], a clipped binary tree protocol is used to resolve collision while in [45, 47] the random access protocol described in [46] is employed.

The second group of priority protocols presented in [48], [49], [50], and [51] allows collisions between high and low priority packets and service priority is given by the collision resolution protocol. In [48], the priority mechanism is implemented by giving a longer back-off time distribution to low-priority packets. In [49], the Part-and-Try algorithm is used for channel random multiple access. The priority access is achieved by allowing a longer transmission permission window for high priority traffic.

In [50] and [51], when a collision between the two types of traffic occurs, the collision between high priority packets is resolved before low priority packets are allowed to attempt retransmissions. In [51], the presence of high priority packets in collision is determined through the use of mini-slots at the beginning of the contention slots. In [50], after the initial collision, low-priority packets are postponed until they detect the end of the collision resolution for the high-priority packets. Both protocols use the random access protocol presented in [46] to resolve collisions. Furthermore, this algorithm allows low-priority stations to detect the moment when the high-priority traffic collision has been resolved.

In order to select or design a random access protocol with mixed priorities, we must recall certain properties inherent to wireless ATM. First, it will operate in a very high speed channel. Second, a wireless ATM frame structure similar to the one illustrated by Figure 3.1 is used. Therefore, feedback via a control slot is not available immediately after each contention slots. Instead, feedback for all uplink control slots in each frame are all transmitted at the same time in the downlink control slots in the next frame. Furthermore, the base station can transmits in the downlink control slots information about the actual state of the multiple access algorithm. Thus a full-sensing algorithm, that requires knowledge of the overall feedback history of the channel in its operation, can

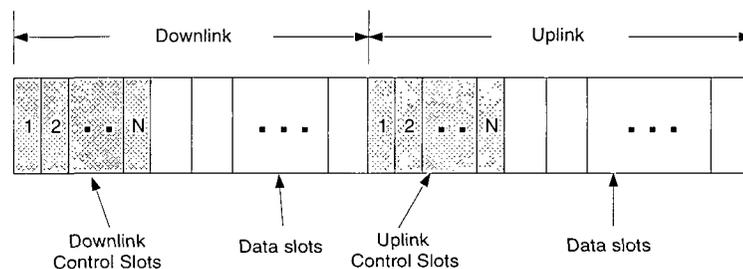


Figure 3.1: Wireless ATM frame structure

be implemented without any problem. One of the design criterion for our MAC protocol is to impose no constraint on the number of users. We can thus make no assumption about the user population size which eliminates all deterministic algorithms. Finally, we must remember that in reservation protocols the overall throughput is not determined by the random access throughput, but the quality of service is highly dependent on the delays encountered by control packets. Hence, we should put more emphasis on the access delay than on the maximum sustainable throughput. We thus want to obtain a relatively constant low delay for high priority traffic in a wide range of total traffic rates without introducing an excessive delay for low-priority traffic.

The protocol presented in [43] can not be selected because it assumes a finite population size to perform its algorithm. Similarly, the protocol described in [42] can not be used since it relies on CSMA and it is known that CSMA presents some pitfalls in the wireless environment [10]. In [43], [44], [47, 45], [50] and [51] the random access algorithm assumes a limited sensing environment, that is each user is required to monitor the channel feedback only from the time it generates a packet and it does not use the previous channel history in its protocol. However, in order to avoid interference with on-going collision resolution, a user with new packets must monitor the channel feedback for a certain time until it detects a certain event (for example, two consecutive non-collision

slots) before attempting a first transmission according to the chosen algorithm. This introduces a fixed overhead to the delay that is unacceptable. These types of random access algorithm must therefore be rejected.

Finally, we are left with two possible protocols: the simple stack algorithm in [48] and the Part-and-Try protocol presented in [49]. The latter protocol clearly outperforms the first one in either the throughput or the delay performance. In fact, the Part-and-Try algorithm is the known random access protocol with the highest throughput (0.487 packets/slot). However, these two protocols, in order to operate correctly, need an immediate feedback after the contention slot, which is not available in wireless ATM. A straightforward modification that can be implemented is to form N different sessions for each uplink control slot in the frame. Then, feedback information transmitted in the downlink control slots can be used to update each independent session. Even if this modification will maintain the maximum global throughput, this separation between sessions is not desirable since it will cause higher delays (for the same reason that N servers of capacity C/N with distinct queues produce a longer delay than a single server of capacity C).

We must thus find a new random access protocol with mixed priorities that may be specifically adapted to the reservation protocol structure and that will meet all the required properties. We have chosen to explore the possibilities offered by the Slotted-Aloha protocol.

3.2 Pseudo-Bayesian Slotted Aloha with Priorities

It is known that the basic slotted Aloha algorithm, where a mobile transmits new packets when it receives them and retransmits backlogged packets with a fixed probability q_r , is unstable for any value of the arrival rate. Thus, to implement a slotted Aloha protocol

with priority classes, we have derived an algorithm similar to the pseudo-Bayesian Aloha stabilization algorithm presented in [2] and [52] and described in the next section. The proposed slotted pseudo-Bayesian priority algorithm is then introduced.

3.2.1 Pseudo-Bayesian Algorithm

In the pseudo-Bayesian algorithm, a new arrival is regarded as backlogged immediately after its arrival and it will attempt transmission in each subsequent slots until success with a probability q_r . Thus, if there are n backlogged packets (including new arrivals) at the beginning of a slot, the attempt rate is $G(n) = nq_r$. The objective of the pseudo-Bayesian algorithm is to maximize the throughput (approximately equal to $G(n)e^{-G(n)}$) by maintaining the attempt rate $G(n)$ equal to one.

The algorithm operates by keeping an updated estimate \hat{n} of the number of backlogged packets n . Each backlogged packet is independently transmitted in a slot t with probability $q_r^t = \min(1, 1/\hat{n}^t)$. Thus, if we have an accurate estimate (i.e. $\hat{n} \approx n$), the attempt rate $G(n)$ will be equal to one. The estimated backlog \hat{n}^{t+1} at the beginning of each slot $t + 1$ is updated from the estimated backlog and feedback for slot t according to the rule:

$$\hat{n}^{t+1} = \begin{cases} \max(\lambda, \hat{n}^t + \lambda - 1) & \text{for idle or success} \\ \hat{n}^t + \lambda + (e - 2)^{-1} & \text{for collision} \end{cases} \quad (3.1)$$

where λ is the estimated arrival rate per slot.

It is known that if the a priori probability distribution of n^t is Poisson with parameter $\hat{n}^t \geq 1$ and if the arrival process is independent of the system state and Poisson with parameter λ , then given an idle or successful transmission in slot t , the probability distribution of n^{t+1} is Poisson with parameter $\hat{n}^t + \lambda - 1$ (in equation (3.1) for the idle or success case, the “maximum” operation is to ensure that if $\hat{n}^t < 1$, the estimated backlog

is at least equal to the number of new arrivals). On the other hand, if there is a collision, the resulting distribution is not quite Poisson but is reasonably approximated as Poisson with parameter \hat{n}^{t+1} .

3.2.2 Slotted Pseudo-Bayesian Priority Algorithm

To implement priority classes, we have derived an algorithm similar to the pseudo-Bayesian algorithm introduced in the previous section. Let new arrivals be regarded as backlogged immediately after their arrival. They will attempt transmission in subsequent slots until success with a probability determined by their priority class and the estimated backlogged state of the system. The cumulative input packet arrival process consists of p independent Poisson processes with intensities $\lambda_1, \dots, \lambda_p$. Let n_1, \dots, n_p and q_1, \dots, q_p , respectively, be the numbers of backlogged packets and the transmission probabilities of the traffic classes. Then, the channel traffic generated by class i is:

$$G_i(n_i) = n_i q_i \quad 1 \leq i \leq p \quad (3.2)$$

and the total attempt rate is:

$$G(n_1, \dots, n_p) = \sum_{i=1}^p G_i(n_i) = \sum_{i=1}^p n_i q_i \quad (3.3)$$

The probability that a packet of the i th traffic class is successfully transmitted in a slot is then given by:

$$\begin{aligned} P_{\text{succ}}^i &= n_i q_i (1 - q_i)^{n_i - 1} \prod_{\substack{j=1 \\ j \neq i}}^p (1 - q_j)^{n_j} \\ &\approx G_i(n_i) e^{-G(n_1, \dots, n_p)} \end{aligned} \quad (3.4)$$

and the probability that a packet from any class is successfully transmitted is:

$$\begin{aligned}
 P_{\text{succ}} &= \sum_{i=1}^p P_{\text{succ}}^i \\
 &= \sum_{i=1}^p \left[n_i q_i (1 - q_i)^{n_i - 1} \prod_{\substack{j=1 \\ j \neq i}}^p (1 - q_j)^{n_j} \right] \\
 &\approx G(n_1, \dots, n_p) e^{-G(n_1, \dots, n_p)}
 \end{aligned} \tag{3.5}$$

We see that if $G(n_1, \dots, n_p)$ is maintained at the optimal value of 1, the system can achieve its maximum throughput of $1/e$. The priority class i throughput will then be $G_i(n_i)/e$. There is thus a possibility to adjust the throughput of each stream to a specific value. Let $\gamma_1, \dots, \gamma_p$ be the priority parameter of each traffic class. We can then relate this priority parameter to each priority class channel traffic as follows:

$$G_i(n_i) = n_i q_i = \gamma_i \quad 1 \leq i \leq p \tag{3.6}$$

If we impose the following constraint on the priority parameters:

$$\sum_{i=1}^p \gamma_i = 1 \tag{3.7}$$

we obtain, using equations (3.3) and (3.5), the desired optimal throughput:

$$P_{\text{succ}} = G_i(n_1, \dots, n_p) e^{-G(n_1, \dots, n_p)} = \left(\sum_{i=1}^p \gamma_i \right) e^{-\sum_{i=1}^p \gamma_i} = 1/e \tag{3.8}$$

and each traffic class i has a throughput of γ_i/e .

Now, assume that before a slot the number n_i of backlogged packets of each priority class i is statistically independent and given by a Poisson distribution with parameter $\hat{n}_i \geq \gamma_i$. We thus have:

$$p(n_i) = \frac{\hat{n}_i^{n_i}}{n_i!} e^{-\hat{n}_i} \tag{3.9}$$

$$p(n_1, \dots, n_p) = \prod_{i=1}^p p(n_i) \tag{3.10}$$

Furthermore, each packet of class i ($1 \leq i \leq p$) is independently transmitted in the next slot with probability $q_i = \gamma_i/\hat{n}_i$ ($\hat{n}_i \geq \gamma_i$) and the priority parameters respect equation (3.7). We will now derive the probability distribution of the number of backlogged packets of each priority traffic class after each of the three possible slot outcomes: idle, success or collision.

Idle Slot Case

We can find the a posteriori joint probability distribution of the number of backlogged packets in the system before a slot given that the slot was idle. We have, using Bayes' rule:

$$p(n_1, \dots, n_p | \text{idle}) = \frac{p(\text{idle} | n_1, \dots, n_p) p(n_1, \dots, n_p)}{p(\text{idle})} \quad (3.11)$$

where:

$$p(\text{idle} | n_1, \dots, n_p) = \prod_{i=1}^p (1 - \gamma_i/\hat{n}_i)^{n_i} \quad (3.12)$$

$$p(\text{idle}) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_p=0}^{\infty} p(\text{idle} | n_1, \dots, n_p) p(n_1, \dots, n_p) = 1/e \quad (3.13)$$

therefore:

$$p(n_1, \dots, n_p | \text{idle}) = e \prod_{i=1}^p \frac{(\hat{n}_i - \gamma_i)^{n_i} e^{-\hat{n}_i}}{n_i!} \quad (3.14)$$

We can then obtain the a posteriori marginal probability distribution of the number of backlogged packets of priority class i ($1 \leq i \leq p$) before a slot, given that the slot was idle:

$$\begin{aligned} p(n_i | \text{idle}) &= \sum_{n_1=0}^{\infty} \cdots \sum_{n_{i-1}=0}^{\infty} \sum_{n_{i+1}=0}^{\infty} \cdots \sum_{n_p=0}^{\infty} p(n_1, \dots, n_p | \text{idle}) \\ &= \frac{(\hat{n}_i - \gamma_i)^{n_i}}{n_i!} e^{-(\hat{n}_i - \gamma_i)} \end{aligned} \quad (3.15)$$

Let n'_i denotes the number of backlogged packets of class i after an idle slot (excluding new arrivals). Then, the n'_i 's joint and marginal probability distributions, given that the slot was idle, are:

$$p(n'_1, \dots, n'_p | \text{idle}) = p(n_1 = n'_1, \dots, n_p = n'_p | \text{idle}) = e \prod_{i=1}^p \frac{(\hat{n}_i - \gamma_i)^{n'_i} e^{-\hat{n}_i}}{n'_i!} \quad (3.16)$$

$$p(n'_i | \text{idle}) = p(n_i = n'_i | \text{idle}) = \frac{(\hat{n}_i - \gamma_i)^{n'_i}}{n'_i!} e^{-(\hat{n}_i - \gamma_i)} \quad (3.17)$$

Thus, excluding new arrivals, the numbers of backlogged packets after an idle slot of each priority class i are independent and Poisson distributed with parameter $\hat{n}_i - \gamma_i$. Furthermore, the arrival process of new packets in the system is Poisson and independent of the contention system. Therefore, the numbers of backlogged packets of each priority class i , including new arrivals, after an idle slot are also independent and Poisson distributed with parameter $\hat{n}_i - \gamma_i + \lambda_i$.

Success Slot Case

Similarly, we can compute the a posteriori joint probability distribution of the number of backlogged packets for each traffic class i ($1 \leq i \leq p$) before a slot, given that in the slot a packet from priority class j ($1 \leq j \leq p$) was successfully transmitted. We have, using Bayes' rule:

$$p(n_1, \dots, n_p | \text{succ}_j) = \frac{p(\text{succ}_j | n_1, \dots, n_p) p(n_1, \dots, n_p)}{p(\text{succ}_j)} \quad (3.18)$$

where:

$$p(\text{succ}_j | n_1, \dots, n_p) = n_j \frac{\gamma_j}{\hat{n}_j} (1 - \gamma_j / \hat{n}_j)^{n_j - 1} \prod_{\substack{i=1 \\ i \neq j}}^p (1 - \gamma_i / \hat{n}_i)^{n_i} \quad (3.19)$$

$$p(\text{succ}_j) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_p=0}^{\infty} p(\text{succ}_j | n_1, \dots, n_p) p(n_1, \dots, n_p) = \gamma_j / e \quad (3.20)$$

therefore:

$$p(n_1, \dots, n_p | \text{succ}_j) = e^{-\frac{(\hat{n}_j - \gamma_j)^{n_j-1} e^{-\hat{n}_j}}{(n_j - 1)!}} \prod_{\substack{i=1 \\ i \neq j}}^p \frac{(\hat{n}_i - \gamma_i)^{n_i} e^{-\hat{n}_i}}{n_i!} \quad (3.21)$$

Using this joint probability distribution, we can find the a posteriori marginal distribution probability of the number of backlogged of priority class j before a slot, given that in the slot a packet from the same priority class j was successful. We then obtain:

$$\begin{aligned} p(n_j | \text{succ}_j) &= \sum_{n_1=0}^{\infty} \cdots \sum_{n_{j-1}=0}^{\infty} \sum_{n_{j+1}=0}^{\infty} \cdots \sum_{n_p=0}^{\infty} p(n_1, \dots, n_p | \text{succ}_j) \\ &= \frac{(\hat{n}_j - \gamma_j)^{(n_j-1)}}{(n_j - 1)!} e^{-(\hat{n}_j - \gamma_j)} \end{aligned} \quad (3.22)$$

We can also find the a posteriori marginal distribution probability, given that in the slot a packet from the priority class j was successfully transmitted, of the number of backlogged of priority class i , $i \neq j$, before the slot:

$$\begin{aligned} p(n_i | \text{succ}_j) &= \sum_{n_1=0}^{\infty} \cdots \sum_{n_{i-1}=0}^{\infty} \sum_{n_{i+1}=0}^{\infty} \cdots \sum_{n_p=0}^{\infty} p(n_1, \dots, n_p | \text{succ}_j) \\ &= \frac{(\hat{n}_i - \gamma_i)^{n_i}}{n_i!} e^{-(\hat{n}_i - \gamma_i)} \quad i \neq j \end{aligned} \quad (3.23)$$

Let n'_i denotes the number of backlogged packets of class i ($1 \leq i \leq p$) after a slot where a packet from priority class j ($1 \leq j \leq p$) was successfully transmitted (excluding new arrivals). Then, the n'_i 's joint and marginal probability distributions are:

$$\begin{aligned} p(n'_1, \dots, n'_p | \text{succ}_j) &= p(n_1 = n'_1, \dots, n_j = n'_j + 1, \dots, n_p = n'_p | \text{succ}_j) \\ &= e \prod_{j=i}^p \frac{(\hat{n}_i - \gamma_i)^{n'_i} e^{-\hat{n}_i}}{n'_i!} \end{aligned} \quad (3.24)$$

$$p(n'_j | \text{succ}_j) = p(n_j = n'_j + 1 | \text{succ}_j) = \frac{(\hat{n}_j - \gamma_j)^{n'_j}}{n'_j!} e^{-(\hat{n}_j - \gamma_j)} \quad (3.25)$$

$$p(n'_i | \text{succ}_j) = p(n_i = n'_i | \text{succ}_j) = \frac{(\hat{n}_i - \gamma_i)^{n'_i}}{n'_i!} e^{-(\hat{n}_i - \gamma_i)} \quad i \neq j \quad (3.26)$$

Thus, excluding new arrivals, the numbers of backlogged packets of each priority class i ($1 \leq i \leq p$) after a slot where a packet from priority class j ($1 \leq j \leq p$) was successfully transmitted are independent and Poisson distributed with parameter $\hat{n}_i - \gamma_i$. Furthermore, the arrival process of new packets in the system is Poisson and independent of the contention system. Thus, the numbers of backlogged packets of each priority class i , including new arrivals, after a success slot are also independent and Poisson distributed with parameter $\hat{n}_i - \gamma_i + \lambda_i$.

Collision Slot Case

Finally, we can compute the a posteriori joint probability distribution of the number of backlogged packets for each traffic class i ($1 \leq j \leq p$) before a slot given that there was a collision in the slot. We then have, using Bayes' rule:

$$p(n_1, \dots, n_p | \text{coll}) = \frac{p(\text{coll} | n_1, \dots, n_p) p(n_1, \dots, n_p)}{p(\text{coll})} \quad (3.27)$$

where:

$$\begin{aligned} p(\text{coll} | n_1, \dots, n_p) &= 1 - p(\text{idle} | n_1, \dots, n_p) - \sum_{i=1}^p p(\text{succ}_i | n_1, \dots, n_p) \\ &= 1 - \prod_{i=1}^p (1 - \gamma_i / \hat{n}_i)^{n_i} \\ &\quad - \sum_{i=1}^p \left[n_i \frac{\gamma_i}{\hat{n}_i} (1 - \gamma_i / \hat{n}_i)^{n_i - 1} \prod_{\substack{j=1 \\ j \neq i}}^p (1 - \gamma_j / \hat{n}_j)^{n_j} \right] \end{aligned} \quad (3.28)$$

$$p(\text{coll}) = 1 - p(\text{idle}) - \sum_{i=1}^p p(\text{succ}_i) = \frac{e - 2}{e} \quad (3.29)$$

therefore:

$$\begin{aligned}
 p(n_1, \dots, n_p | \text{coll}) &= \frac{e}{e-2} \prod_{i=1}^p \frac{e^{-\hat{n}_i}}{n_i!} \left(\prod_{i=1}^p \hat{n}_i^{n_i} - \prod_{i=1}^p (\hat{n}_i - \gamma_i)^{n_i} \right. \\
 &\quad \left. - \sum_{i=1}^p \left[n_i \gamma_i (\hat{n}_i - \gamma_i)^{n_i-1} \prod_{\substack{j=1 \\ j \neq i}}^p (\hat{n}_j - \gamma_j)^{n_j} \right] \right) \quad (3.30)
 \end{aligned}$$

Using this joint probability distribution, we can find the a posteriori marginal distribution probability of the number of backlogged of priority class i ($1 \leq i \leq p$) before a slot, given that in the slot a collision occurred. We then obtain:

$$\begin{aligned}
 p(n_i | \text{coll}) &= \sum_{n_1=0}^{\infty} \cdots \sum_{n_{i-1}=0}^{\infty} \sum_{n_{i+1}=0}^{\infty} \cdots \sum_{n_p=0}^{\infty} p(n_1, \dots, n_p | \text{coll}) \\
 &= \frac{e}{e-2} \frac{e^{-\hat{n}_i}}{n_i!} \left[\hat{n}_i^{n_i} - e^{\gamma_i-1} \left[(2 - \gamma_i) (\hat{n}_i - \gamma_i)^{n_i} \right. \right. \\
 &\quad \left. \left. + n_i \gamma_i (\hat{n}_i - \gamma_i)^{n_i-1} \right] \right] \quad (3.31)
 \end{aligned}$$

Let n'_i denote the number of backlogged packets of class i ($1 \leq i \leq p$) after a slot where a collision occurred (excluding new arrivals). Then, the n'_i 's joint and marginal probability distributions after a collision slot are:

$$\begin{aligned}
 p(n'_1, \dots, n'_p | \text{coll}) &= p(n_1 = n'_1, \dots, n_p = n'_p | \text{coll}) \\
 &= \frac{e}{e-2} \prod_{i=1}^p \frac{e^{-\hat{n}_i}}{n'_i!} \left(\prod_{i=1}^p \hat{n}_i^{n'_i} - \prod_{i=1}^p (\hat{n}_i - \gamma_i)^{n'_i} \right. \\
 &\quad \left. - \sum_{i=1}^p \left[n'_i \gamma_i (\hat{n}_i - \gamma_i)^{n'_i-1} \prod_{\substack{j=1 \\ j \neq i}}^p (\hat{n}_j - \gamma_j)^{n'_j} \right] \right) \quad (3.32)
 \end{aligned}$$

$$\begin{aligned}
 p(n'_i | \text{coll}) &= p(n_i = n'_i | \text{coll}) \\
 &= \frac{e}{e-2} \frac{e^{-\hat{n}_i}}{n'_i!} \left[\hat{n}_i^{n'_i} - e^{\gamma_i-1} \left[(2 - \gamma_i) (\hat{n}_i - \gamma_i)^{n'_i} \right. \right. \\
 &\quad \left. \left. + n'_i \gamma_i (\hat{n}_i - \gamma_i)^{n'_i-1} \right] \right] \quad (3.33)
 \end{aligned}$$

We can clearly see that after a collision slot, the number of backlogged packets in each priority class i are not independent nor Poisson distributed. However, we can compute the mean and variance of the obtained marginal distribution and compare with the mean and variance of a random variable X that is Poisson distributed with parameter $\hat{n}_i + \gamma_i/(e - 2)$. We then find:

$$E[n'_i | \text{coll}] = \hat{n}_i + \frac{\gamma_i}{e - 2} \qquad E[x] = \hat{n}_i + \frac{\gamma_i}{e - 2} \qquad (3.34)$$

$$\text{Var}[n'_i | \text{coll}] = \hat{n}_i + \frac{\gamma_i}{e - 2} - \left(\frac{\gamma_i}{e - 2} \right)^2 \qquad \text{Var}[x] = \hat{n}_i + \frac{\gamma_i}{e - 2} \qquad (3.35)$$

We thus see that even if the distribution of n'_i , given that there was a collision, is not quite Poisson, its mean and variance are similar to the Poisson distribution with parameter $\hat{n}_i + \gamma_i/(e - 2)$. Furthermore, we know that $\gamma_i \leq 1$ and that \hat{n}_i is likely to be large when a collision occurs. Under these conditions, the variance becomes almost equal to the variance of a Poisson distribution. The similarity of the two distributions can also be observed through plotting of their respective probability density function. For these reasons, we can say that the number of backlogged packets for each priority class i ($1 \leq i \leq p$) after a collision slot, including new arrivals, is reasonably approximated as Poisson with parameter $\hat{n}_i + \gamma_i/(e - 2) + \lambda_i$.

The distributions are still not independent, but we can compute the correlation matrix to find an indication of the degree of dependency between the variable. We find that the correlation between the number of backlogged packets of two different priority classes i and j ($i \neq j$) is given by:

$$\begin{aligned} \text{Corr}[n'_i, n'_j | \text{coll}] &= \frac{E[n'_i n'_j | \text{coll}] - E[n'_i | \text{coll}] E[n'_j | \text{coll}]}{\sqrt{\text{Var}[n'_i | \text{coll}] \text{Var}[n'_j | \text{coll}]}} \\ &= \frac{-\gamma_i \gamma_j}{(e - 2)^2 \left[\left(\hat{n}_i + \frac{\gamma_i}{e - 2} - \left(\frac{\gamma_i}{e - 2} \right)^2 \right) \left(\hat{n}_j + \frac{\gamma_j}{e - 2} - \left(\frac{\gamma_j}{e - 2} \right)^2 \right) \right]^{1/2}} \end{aligned} \qquad (3.36)$$

Since $\gamma \leq 1$ and \hat{n} 's are likely to be large when there is a collision, we see that the correlation between the number of backlogged packets of different priority classes is quite small. Furthermore, the arrival streams for each traffic class are independent. Therefore, we can reasonably assume that the n_i 's are independent.

Algorithm

In the previous analysis, we showed that our initial assumptions on the independent and Poisson distributions of the number of backlogged packet of each traffic class i ($1 \leq i \leq p$) are satisfied for the three possible slot outcomes: idle, success or collision slot. We can therefore, based on these results, derive an algorithm similar to the pseudo-Bayesian algorithm to implement a multiple access protocol with mixed priorities. Suppose that we have p different priority classes with independent Poisson arrival processes of intensities $\lambda_1, \dots, \lambda_p$. A lower index corresponds to a higher priority packet class. Let γ_i be the priority parameter of each traffic class i . To maintain the priority order, we must have $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_{p-1} \geq \gamma_p$ and the parameters satisfy the relation $\sum_{i=1}^p \gamma_i = 1$.

The algorithm operates by maintaining for each priority class i an estimate \hat{n}_i^t of the number of backlogged packets n_i^t at the beginning of each slot t . For each priority class i , an effective priority parameter $\hat{\gamma}_i^t$ is also computed (used to avoid that $\gamma_i \geq \hat{n}_i^t$). A new arrival during slot t is immediately regarded as backlogged and it will attempt transmission in each subsequent slot after its arrival until success.

At the beginning of each slot t , \hat{n}_i^t is updated from \hat{n}_i^{t-1} , $\hat{\gamma}_i^{t-1}$ and the feedback for slot $t-1$ according to the rule:

$$\hat{n}_i^t = \begin{cases} \max(\lambda_i, \hat{n}_i^{t-1} + \lambda_i - \hat{\gamma}_i^{t-1}) & \text{for idle or success} \\ \hat{n}_i^{t-1} + \lambda_i + \frac{\hat{\gamma}_i^{t-1}}{e-2} & \text{for collision} \end{cases} \quad (3.37)$$

The priority parameters γ_i are fixed values set when the system is initialized. However, the transmission probabilities for each slots t are given by γ_i/\hat{n}_i^t and they can not

be greater than one. Therefore, if for one or more priority classes i we have $\gamma_i > \hat{n}_i^t$, then $q_i^t = 1$ and then the “effective” values of the priority parameters γ_i for the throughput equations are no longer equal to their initial values but to \hat{n}_i^t (see equation (3.6)). Furthermore, since $\hat{n}_i^t < \gamma_i$, we have $\sum_{i=1}^p \gamma_i < 1$ and the total throughput is lower than its optimal value of $1/e$ (equation (3.8)). Thus, to maintain the optimal throughput, we should assign the difference between the fixed γ_i and \hat{n}_i^t to another priority class. We therefore use a prorating algorithm to dynamically compute, at the beginning of each frame t , for each priority class i , the effective priority parameter $\hat{\gamma}_i^t$ based on the fixed priority parameters γ_i and the estimated numbers of backlogged packets \hat{n}_i^t .

The parameters $\hat{\gamma}_i^t$ are set such that $\sum_{i=1}^p \hat{\gamma}_i^t = 1$ and, if $\sum_{i=1}^p \hat{n}_i^t \geq 1$, then $\hat{\gamma}_i \leq \hat{n}_i^t$ for all priority classes. The prorating algorithm ensures that for each priority class i , its effective priority parameter $\hat{\gamma}_i^t$ is set to γ_i if $\gamma_i \leq \hat{n}_i^t$, or \hat{n}_i^t , otherwise. If $\sum_{i=1}^p \hat{\gamma}_i^t < 1$, to maintain the optimum throughput, the “leftover” (i.e. $1 - \sum_{i=1}^p \hat{\gamma}_i^t$) is added to the effective priority parameter of the highest priority class (i.e. $\hat{\gamma}_1^t$) in order to increase its transmission probability and therefore decrease its waiting time. Then, if $\hat{\gamma}_1^t \leq \hat{n}_1^t$ the prorating algorithm is stopped, otherwise, $\hat{\gamma}_1^t$ is set to \hat{n}_1^t and the same procedure is repeated for each priority class i ($i > 1$) in order of decreasing priority. After this procedure, if $\sum_{i=1}^p \hat{\gamma}_i^t < 1$, the “leftover” will be assign to each priority class proportionally to their arrival rate. The prorating algorithm can be summarized as follows:

for (each priority class i in order of increasing priority)

$$\hat{\gamma}_i^t = \min\left(\hat{n}_i^t, 1 - \sum_{j=1}^{i-1} \hat{\gamma}_j^t - \sum_{j=i+1}^p \min(\hat{n}_j^t, \gamma_j)\right)$$

$$L = 1 - \sum_{i=1}^p \hat{\gamma}_i^t$$

for (each priority class i)

$$\hat{\gamma}_i^t = \hat{\gamma}_i^t + \frac{\lambda_i}{\sum_{j=1}^p \lambda_j} L$$

Then each backlogged packet is independently transmitted in slot t according to the transmission probability q_i^t of the priority class i it belongs to. Transmission probabilities are calculated as follows:

$$q_i^t = \min\left(1, \frac{\hat{\gamma}_i^t}{\hat{n}_i^t}\right) \quad (3.38)$$

3.3 Framed Pseudo-Bayesian Aloha with Priorities

In order to work in wireless ATM environment, we must adapt the slotted pseudo-Bayesian priority algorithm to the framed structure of reservation protocols. The pseudo-Bayesian protocols introduced until now work on a slot basis and require an immediate feedback after the contention slot. However, in reservation protocols time slots are grouped into frames and feedback is available only once per frame. In the next section we introduce a strategy to adapt the original pseudo-Bayesian algorithm to the framed environment. Then, the framed pseudo-Bayesian priority algorithm is described.

3.3.1 Framed Pseudo-Bayesian Algorithm

The straightforward modification to adapt the slotted pseudo-Bayesian algorithm to the frame structure is to create K different sessions for each slot k ($1 \leq k \leq K$) in the frame. However, as explained in Section 3.1, this scheme will introduce longer delays. Instead, we propose a strategy where a packet wait until the end of the on-going frame before attempting its first transmission (gated system). Starting from the next frame, it will independently attempt to transmit with a probability q_i in a randomly chosen slot for each frame. The selected slot is independent from frame to frame.

Let suppose that there is K slots in a frame, the arrival process is Poisson with parameter λ (arrival rate per frame) and the a priori distribution of the total number

of backlogged packets n^t at the beginning of frame t is Poisson with parameter \hat{n}^t . If each backlogged packets chooses independently a given slot in the frame for transmission with an uniform probability (each slot has a probability $1/K$ of being chosen), then the probability distribution of the number of backlogged packets n_k^t that chose a given slot k is:

$$\begin{aligned}
 p(n_k^t) &= \sum_{n^t=0}^{\infty} p(n_k^t | n^t) p(n^t) \\
 &= \sum_{n^t=n_k^t}^{\infty} \binom{n^t}{n_k^t} \left(\frac{1}{K}\right)^{n_k^t} \left(1 - \frac{1}{K}\right)^{n^t-n_k^t} \frac{(\hat{n}^t)^{n^t}}{n^t!} e^{-\hat{n}^t} \\
 &= \frac{\left(\frac{\hat{n}^t}{K}\right)^{n_k^t}}{n_k^t!} e^{-\left(\frac{\hat{n}^t}{K}\right)}
 \end{aligned} \tag{3.39}$$

Therefore, n_k^t , the number of backlogged packets for each slot k ($1 \leq k \leq K$) at the beginning of frame t , is Poisson with parameter \hat{n}^t/K and it can be shown that they are independent for each slot k [53]. We can then apply the pseudo-Bayesian algorithm presented in Section 3.2.1 independently for each slot. The moment of the feedback is not important, as long as we receive it before the next frame. Therefore, \hat{n}_k^{t+1} , the updated estimated number of backlogged packets in a slot k after frame t , can be computed with equation (3.1) and n_k^{t+1} , the numbers of backlogged packets in each slot k at the end of frame t , are independent Poisson random variables with parameter \hat{n}_k^{t+1} . Then n^{t+1} , the total number of backlogged packets at the beginning of frame $t+1$, is given by:

$$n^{t+1} = \sum_{k=0}^K n_k^{t+1} \tag{3.40}$$

Since the n_k^{t+1} 's are independent Poisson random variables, n^{t+1} is also Poisson with mean $\sum_{k=0}^K \hat{n}_k^{t+1}$.

Algorithm

Using these results, we propose the following slotted pseudo-Bayesian algorithm for a K slots frame and where the arrival rate λ is given in number of packets per frame. The algorithm operates by maintaining an updated estimate \hat{n} of the total number of backlogged packets in the system. A new arrival during frame t is immediately regarded as backlogged and it will independently attempt transmission in each subsequent frame after its arrival (i.e. starting with frame $t + 1$) until success. Each backlogged packet is independently transmitted in a frame t with probability $q_r^t = \min(1, K/\hat{n}^t)$ over a slot independently chosen with a uniform probability (each slot has a probability $1/K$ of being chosen). Therefore, if we have an accurate estimate (i.e. $\hat{n} \approx n$), then the attempt rate per slot $G(n) = \frac{n}{K}q_r$ will be equal to the optimal value of one.

For each frame t , the estimated total number of backlogged packets at the beginning of frame $t + 1$ is updated from the estimated total backlog at the beginning of frame t and feedback from each slot k ($1 \leq k \leq K$) of frame t (let k_{nc} be the number of idle or success slots and k_c the number of collision slots in frame t) according to the rule:

$$\hat{n}^{t+1} = \lambda + k_{nc} \max\left(0, \frac{\hat{n}^t}{K} - 1\right) + k_c \left(\frac{\hat{n}^t}{K} + \frac{1}{e-2}\right) \quad (3.41)$$

Algorithm Evaluation

The attempt rate per slot in the new scheme is $G(n) = \frac{n}{K}q_r$ and the probability of success in a given slot (i.e. the throughput) is $P_{\text{succ}} \approx G(n)e^{-G(n)}$. Thus the maximum achievable throughput per slot is e^{-1} when $G(n) = 1$, which is the objective that the framed pseudo-Bayesian algorithm tries to achieve. Since each individual slot independently obtains a throughput of e^{-1} then the total throughput is still e^{-1} . Our scheme uses the same algorithm as the normal pseudo-Bayesian, it is thus stable under the same conditions (i.e. if a fixed λ value of e^{-1} is used within the algorithm, stability is achieved

for all actual arrival rate smaller than e^{-1} [2, 52]). In our simulations, we have used a dynamic estimate of λ computed by averaging the number of success over a sliding time window of 500 slots. Nothing has been proved about the stability of such a scheme; however, in our simulations, no instability has been observed.

To evaluate the delay performance of our scheme, we will compare our results with the performance of protocols where K different sessions are formed for each of the N frame slots. We used a modified version of the *Time-Division Multiplexing* (TDM) system and of the pseudo-Bayesian slotted Aloha algorithm. We take a slot as the time unit and the total arrival rate λ is given in number of packets per slot.

In the TDM system, we have K traffic streams that are time-division multiplexed in a scheme where the time axis is divided in K -slot frames with one slot dedicated to each traffic stream. Packets arrive according to a Poisson process with rate λ/K for each stream. The system is gated, that is a new packet can join the queue only in the next frame following its arrival. We can see the scheme for each slot in the frame as a system where transmissions can only start at the beginning of a frame and the service time is K time units. We thus have an $M/D/1$ queuing system with vacations where $\mu = 1/K$, $\rho = \lambda$, $\bar{V} = K$ and $\bar{V}^2 = K^2$. This give us the waiting time from the arrival of a packet to the beginning of the frame where it will be transmitted. We then have an average additional waiting time of $(K - 1)/2$ before the packet can be transmitted in the slot associated with the session that the packet belongs to. Thus, using results from [2], the total waiting time before transmission for this system is:

$$W_{TDM} = K - \frac{1}{2} + \frac{\lambda K}{2(1 - \lambda)} \quad (3.42)$$

We can apply a similar analysis for the slotted Aloha system where a new arrival chooses a slot in the frame at its arrival and only contend in this slot starting from the frame following its arrival. We then find, using results presented in [2], that the waiting

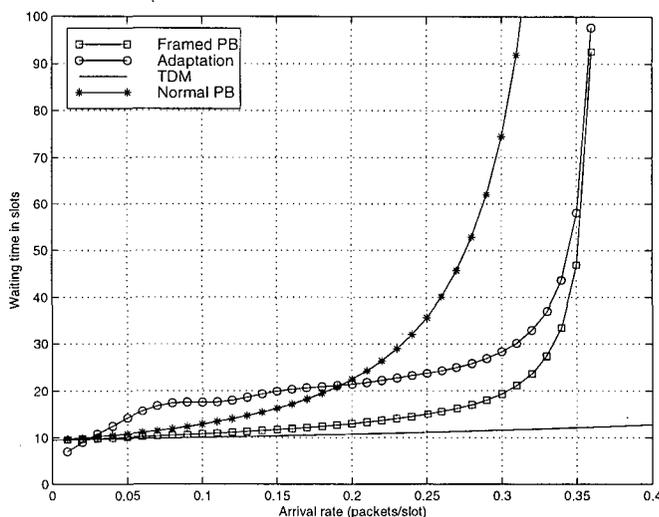


Figure 3.2: Waiting time in number of slots

time is:

$$W_{SA} = K \left[\frac{e - 1/2}{1 - \lambda e} - \frac{(e^\lambda - 1)(e - 1)}{\lambda[1 - (e - 1)(e^\lambda - 1)]} \right] + \frac{K - 1}{2} \quad (3.43)$$

We have also evaluated another adaptation scheme derived from the framed pseudo-Bayesian algorithm where the number of slots in a frame is dynamically adjusted as a function of the total estimated number of backlogged packets. In this modified protocol, K^t , the number of slots in a frame t , is set to $\lceil \hat{n}^t \rceil$. However, the frame length remains equal to K ; only K^t slots in frame t are used for contention and the other slots are idle (in a real system they can be used for other purposes).

Figure 3.2 shows the results of the simulations for the two proposed systems compared to the theoretical waiting time of the two reference schemes with a frame length of 10 slots (i.e. $K = 10$). We see from the results that our proposed schemes outperform the straightforward implementation of slotted Aloha in a framed environment. This confirms our hypothesis that the implementation of independent collision resolution sessions for

each slot in a frame reduces the delay performance of the scheme. If we compare our schemes with the TDM system, we see that for arrival rates lower than 0.3 packets/slot the delay experienced by each packets is in the same order of magnitude. In fact, the contention delay difference between the framed pseudo-Bayesian algorithm and the TDM system (the perfect multiplexing system) is less than 10 slots in this region which is less than a frame.

If we compare the two proposed schemes, we see that the delay for the adaptation scheme is approximately 10 slots (or 1 frame) greater than for the fixed scheme. This is due to the fact that when there is a burst of arrivals, there is a latency (at least one frame) before the frame reaches its optimal size. For low arrival rate, the delay is shorter for the adaptation scheme because the useful slots are taken at the beginning of the frame such that the chosen transmission slot is before the average one for the fixed system.

In a real wireless ATM structure, the uplink contention control slots are grouped in a short period of time and are not spanned over the entire frame. This justifies the use of a gated system. For the same reason, we are more interested in the frame delay than in the slot delay. The frame delay represents the number of frames that a packets must wait before its transmission (i.e. departure frame - arrival frame). Figure 3.3 shows the average waiting time in number of frames for the two proposed schemes. These simulation results confirm the one frame latency of the adaptation scheme compared to the fixed system. What is interesting is that for arrival rates smaller than 0.3, the average delay for the proposed fixed framed pseudo-Bayesian algorithm is lower than 2 frames. This is excellent since, due to the gated nature of the system, there is a fixed minimal waiting time of one frame. Finally, Figure 3.4 shows the average number of slots used for contention by the adaptation algorithm. Thus, if the number of slots per frame assigned to control slots is an important issue, the adaptation algorithm can provide an advantage. However, the price to pay is a one frame increase in the average packet delay.

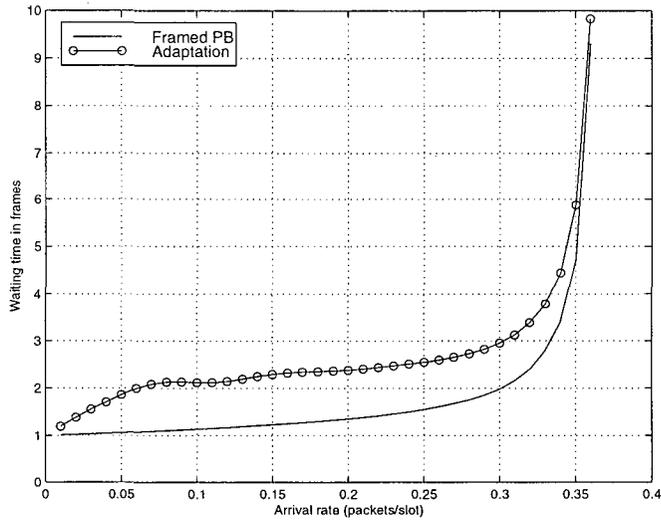


Figure 3.3: Waiting time in number of frames

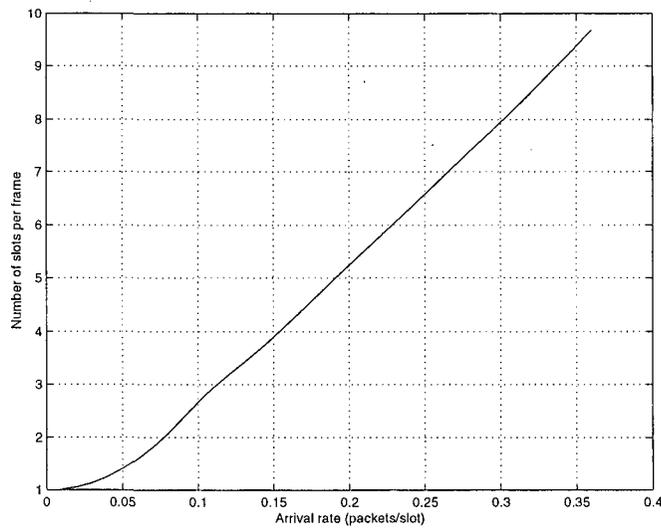


Figure 3.4: Average number of contention slots per frame

3.3.2 Framed Pseudo-Bayesian Priority Algorithm

Using the results from Section 3.3.1, we can derive from the slotted pseudo-Bayesian priority algorithm presented in Section 3.2.2 a multiple access protocol with mixed priorities for a K slots frame. The arrival rate λ_i for each priority class i ($1 \leq i \leq p$) is given in number of packets per frame. The same definitions that were presented in Section 3.2.2 for γ , $\hat{\gamma}$ and priority order are assumed.

The algorithm operates by maintaining for each priority class i an estimate \hat{n}_i^t of the total number of backlogged packets n_i^t at the beginning of each frame t . A new arrival during frame t is immediately regarded as backlogged and it will attempt transmission in each subsequent frame after its arrival until success.

Similarly to what has been shown in equation (3.39), we can show that if each of the \hat{n}_i^t backlogged packet of priority class i chooses independently one of the K slots in the frame for transmission with an uniform probability, then at the beginning of frame t , the number of backlogged packets of priority class i for each slot k ($1 \leq k \leq K$) is Poisson and independent with parameter \hat{n}_i^t/K . We can then independently apply for each slot k the pseudo-Bayesian rule given by equation (3.37) to compute the updated estimate of the number of backlogged packets after the slot. Furthermore, according to the results in Section 3.3.1, \hat{n}_i^{t+1} , the updated estimate of the total number of backlogged packets of priority class i at the end of frame t , is given by the sum of the updated estimate of each slot k in frame t . Therefore, at the beginning of each frame t , for each priority class i , \hat{n}_i^t is updated from \hat{n}_i^{t-1} , $\hat{\gamma}_i^{t-1}$ and the feedback for frame $t-1$ (let k_{nc} be the number of idle or success slots and k_c the number of collision slots in frame $t-1$) according to the rule:

$$\hat{n}_i^t = \lambda_i + k_{nc} \max\left(0, \frac{\hat{n}_i^{t-1}}{K} - \hat{\gamma}_i^{t-1}\right) + k_c \left(\frac{\hat{n}_i^{t-1}}{K} + \frac{\hat{\gamma}_i^{t-1}}{e-2}\right) \quad (3.44)$$

For the prorating algorithm, the objective is to maintain the optimum throughput

in each slot k in the frame. We can therefore use the prorating algorithm presented in Section 3.2 with the only difference that the estimated number of backlogged packet in a slot is given by \hat{n}_i^t/K . We thus find the effective priority parameter of each priority class using the following prorating algorithm:

for (each priority class i in order of increasing priority)

$$\hat{\gamma}_i^t = \min\left(\frac{\hat{n}_i^t}{K}, 1 - \sum_{j=1}^{i-1} \hat{\gamma}_j^t - \sum_{j=i+1}^p \min\left(\frac{\hat{n}_j^t}{K}, \gamma_j\right)\right)$$

$$L = 1 - \sum_{i=1}^p \hat{\gamma}_i^t$$

for (each priority class i)

$$\hat{\gamma}_i^t = \hat{\gamma}_i^t + \frac{\lambda_i}{\sum_{j=1}^p \lambda_j} L$$

Then each backlogged packet is independently transmitted in frame t according to the transmission probability q_i^t of the priority class i it belongs to. The transmission probability for each class i is selected such that the attempting rate in each slot of the frame is maintained to its optimal value. We can therefore use equation (3.38) to compute the transmission probabilities by replacing the estimated number of backlogged packet by \hat{n}_i^t/K . Transmission probabilities are thus calculated as follows:

$$q_i^t = \min\left(1, \frac{\hat{\gamma}_i^t}{\hat{n}_i^t} K\right) \quad (3.45)$$

If a packet is transmitted in a given frame, it will independently choose a contention slot for transmission with an uniform probability (each slot has a probability $1/K$ of being chosen).

3.4 Simulation Results with Poisson Traffic

A C++ simulator has been written to evaluate the performance of the proposed systems under various conditions. Representative results are presented in this section. Three

different systems have been examined:

- *Slotted Pseudo-Bayesian Priority* (SPBP) algorithm (Section 3.2.2);
- *Framed Pseudo-Bayesian Priority* (FPBP) algorithm (Section 3.3.2);
- *Framed Adaptation Pseudo-Bayesian Priority* (FAPBP) algorithm.

The last system is a modification of the framed pseudo-Bayesian priority algorithm proposed in Section 3.3.2 similar to the modification presented in Section 3.3.1. In this adaptation algorithm, the number of slots in a frame remains fixed; however, in frame t , only K^t slots among the K slots can be used for contention where K^t is set equal to $\lceil \sum_{i=1}^p \hat{n}^i \rceil$.

To evaluate the proposed systems, the delay measurements have been compared with results obtained with the non-priority algorithms. The slotted pseudo-Bayesian, the framed pseudo-Bayesian and the framed adaptation pseudo-Bayesian priority algorithms are respectively compared with the systems presented in Section 3.2.1, 3.3.1 and 3.3.1. The arrival rate value used by the backlogged estimation algorithm is an estimate computed with the moving time-average of successful transmissions for each traffic class over a window period of 500 slots. Simulations have also been run with a perfect knowledge of the arrival rate. However, results for these simulations are not presented since they are similar to those obtained with the moving average estimation and also because in a real system no a priori knowledge of the arrival rate is available. Furthermore, the moving average estimation algorithm gives slightly better results since it has the possibility to adapt to short term variation of the arrival rate. We have simulated each experimental condition for a period of at least 10 million slots. We do not have any confidence interval for the results; however the length of the simulations ensures the validity of the results. The framed schemes have been simulated with ten slots per frame and delays

calculated in number of frames are presented for this scheme. Only delays calculated in number of frames are presented for the two framed algorithms since, as explained in Section 3.3.1, the waiting time in number of slots has no real signification in the wireless ATM environment

Before presenting the results, we recall the required objectives that our new schemes should fulfill:

- Low and stable waiting time for high priority traffic over a wide range of total arrival rates;
- “Average” results similar to the reference algorithm;
- Small degradation of low-priority traffic delay.

We have performed simulations to completely evaluate the performance of the schemes with two different priority classes. We then simulated other experimental conditions to evaluate the performance of the pseudo-Bayesian priority algorithms with more than two priority classes. Simulation results illustrating the impact of the FPBP algorithm on the DR-TDMA protocol performance are presented in Chapter 4 (Section 4.4.3).

3.4.1 Performance with Two Priority Classes

In this section we have simulated the priority algorithms with two different priority classes. The total arrival process is Poisson with intensity $\lambda_1 + \lambda_2$ and each packet has an independent probability $\lambda_1/(\lambda_1 + \lambda_2)$ of being part of the traffic class one, otherwise it is a traffic class two packet. On each figure, four different results are presented: statistics for class one, statistics for class two, results obtained by considering class one and class two as a single population in statistics computations (“Average” line), and the reference algorithm (“Basic PB” line).

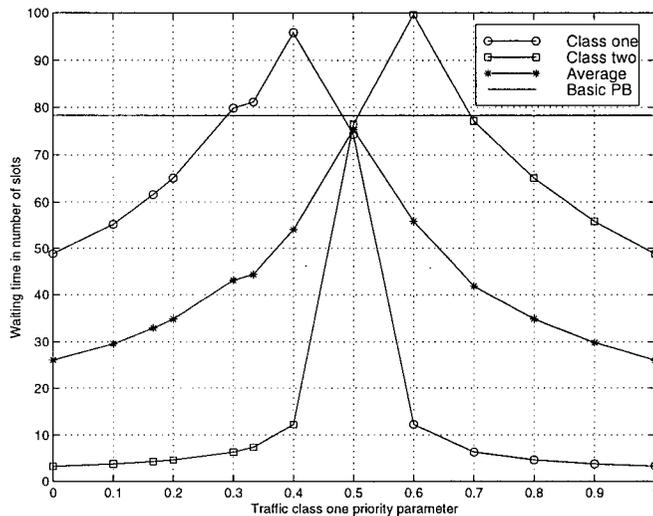


Figure 3.5: Waiting time as a function of the priority parameter for the SPBP system. $\lambda_1 = 0.18$ and $\lambda_2 = 0.18$ packets/slot.

Figures 3.5 to 3.8 show the effect of the priority parameter on the waiting time for a fixed arrival rate. From the throughput equations presented in Section 3.2.2, we see that a traffic class i has a portion γ_i of the total throughput. Thus, we should expect that when a traffic class has an arrival rate smaller than its share of the total throughput (i.e. $\lambda_i \leq \gamma_i/e$), it will experience lower delays than the other traffic classes. We can clearly observe this phenomenon from the results: each traffic class has the same delay when their arrival rate is equivalent to their share of the total throughput, that is, when $\gamma_i = \lambda_i / \sum_{j=1}^p \lambda_j$. We should also underline the sharp “cutoff”. We can thus select the priority parameter without having a good a priori knowledge of the traffic characteristics of each class. Finally, there is an interesting phenomenon that we can observe in the figures: the delay experienced by each traffic class (both lower and higher priority) decreases as we move away from the fair share point. That is, the more we favor

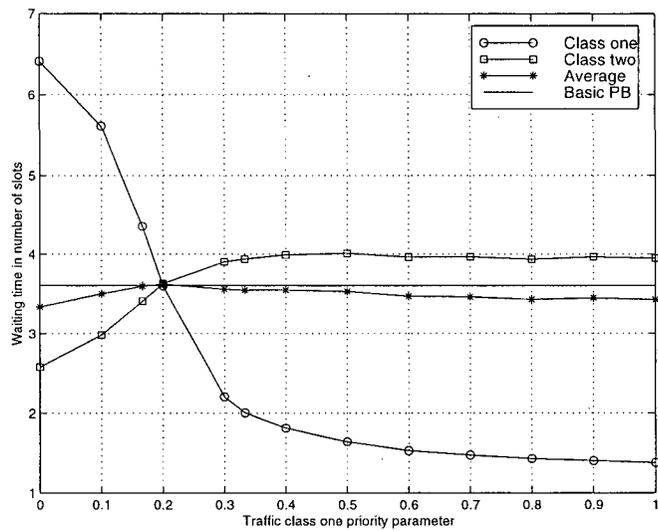


Figure 3.6: Waiting time as a function of the priority parameter for the SPBP system. $\lambda_1 = 0.05$ and $\lambda_2 = 0.20$ packets/slot.

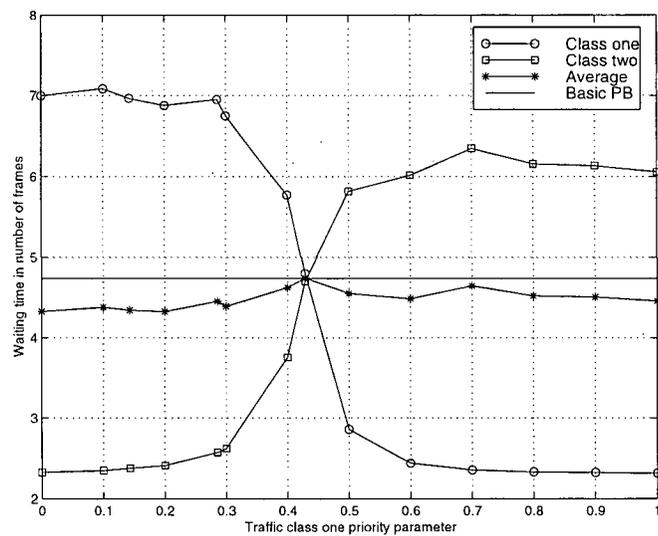


Figure 3.7: Waiting time as a function of the priority parameter for the FPBP system. $\lambda_1 = 0.15$ and $\lambda_2 = 0.20$ packets/slot.

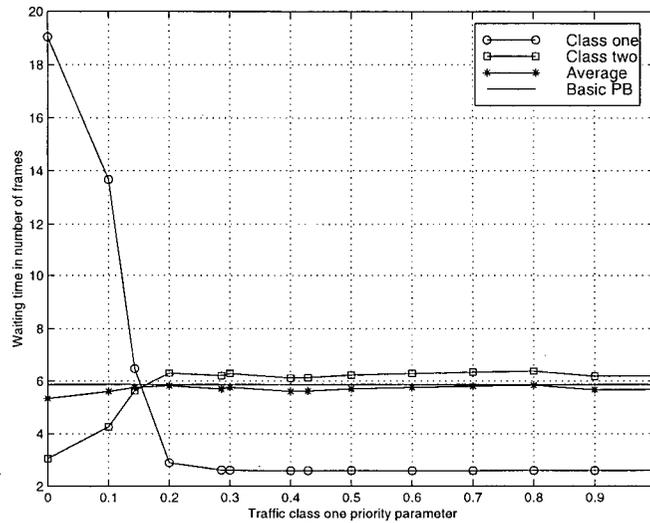


Figure 3.8: Waiting time as a function of the priority parameter for the FAPBP system. $\lambda_1 = 0.05$ and $\lambda_2 = 0.30$ packets/slot.

a traffic class, not only does its delay decrease, as we expect, but the delay of the low-priority class also decreases. This phenomenon can be explained by the fact that if we give a large priority parameter to a traffic class, there are less packets allowed to contend for the channel and there are less collisions, thus reducing the high priority delay. The prorating algorithm helps the low priority packets to take advantage of the high priority delay reduction and to decrease the low-priority delay. Thus, the optimum operating point is at $\gamma_1 = 1$ (if we want to give priority to class one traffic).

The waiting time as a function of the high priority arrival rate is illustrated in Figures 3.9 to 3.12 for the three systems. Since, as explained previously, the optimal operating point is at $\gamma_1 = 1$, only the results with this condition are presented. We have also simulated the systems for $\gamma_1 = 0.5$ and $\gamma_1 = 0.75$. However, the results are not interesting since the performance is better with $\gamma_1 = 1$ for both the waiting times of the low-priority and high-priority packets. From these result, we can observe that the

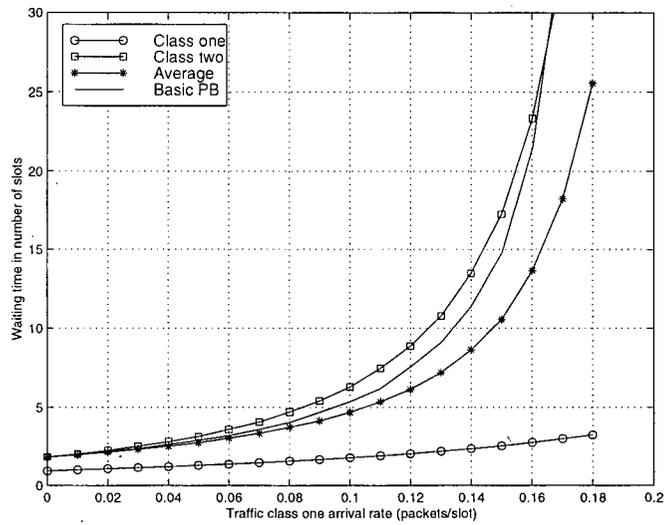


Figure 3.9: Waiting time as a function of traffic class one arrival rate for the SPBP system. $\lambda_2 = 0.18$ packets/slot and $\gamma_1 = 1$.

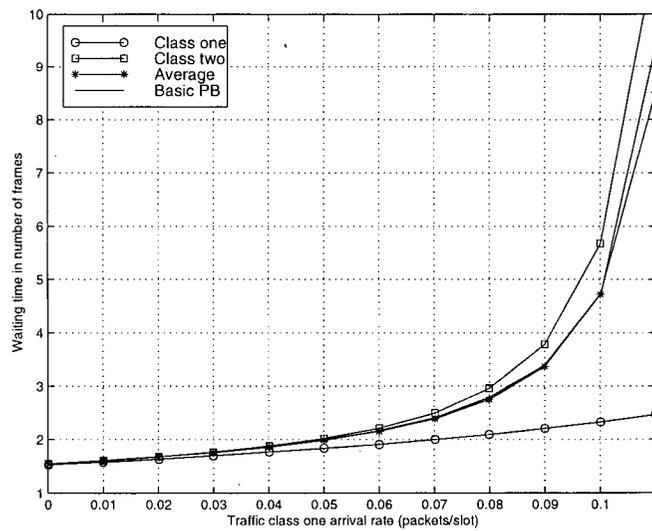


Figure 3.10: Waiting time as a function of traffic class one arrival rate for the FPBP system. $\lambda_2 = 0.25$ packets/slot and $\gamma_1 = 1$.

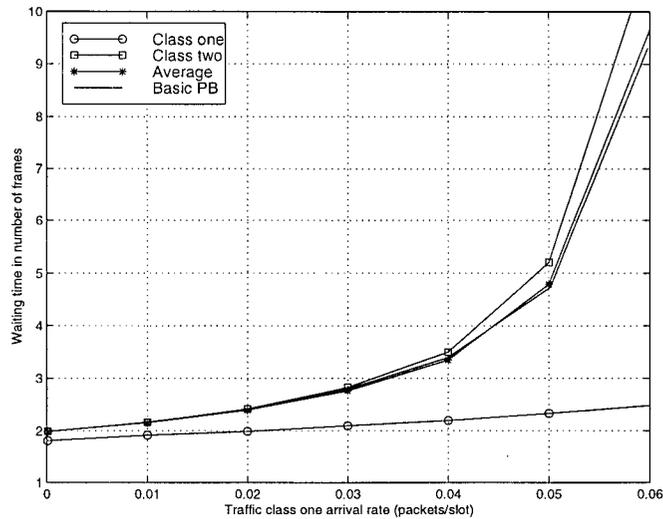


Figure 3.11: Waiting time as a function of traffic class one arrival rate for the FPBP system. $\lambda_2 = 0.30$ packets/slot and $\gamma_1 = 1$.

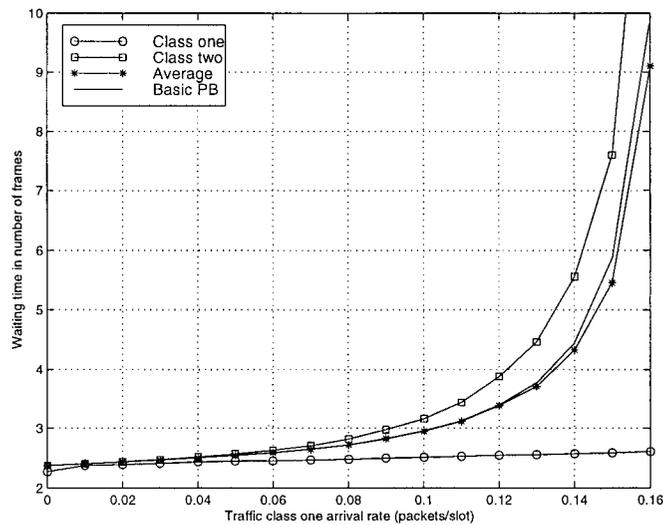


Figure 3.12: Waiting time as a function of traffic class one arrival rate for the FAPBP system. $\lambda_2 = 0.20$ packets/slot and $\gamma_1 = 1$.

proposed protocols meet our requirement, that is:

- The high-priority waiting time is stable for a large range of traffic conditions. Furthermore, even near the maximal total arrival rate (0.368 packets/slot) the waiting time of high priority packets remains almost at the same level;
- The average waiting time of the two classes together is always lower than that of the reference protocol;
- The low-priority traffic class suffers only a small degradation of its waiting time compare to the reference.

We also see that regardless of the intensity of the low priority traffic (from $\lambda_2 = 0.18$ to $\lambda_2 = 0.3$ packets/slot), the maximal average waiting time for the high-priority traffic class stays under 4 slots for the SPBP system and 3 frames for both the FPBP and FAPBP systems even if the total arrival rate is equal to 0.368 packets/slot. In fact, from the throughput equations presented in Section 3.2.2, we can deduce that the sub-system composed of traffic class one will remain stable while its arrival rate stays under $1/e$. For the general case, even if the total throughput is limited to $1/e$, the throughput of an individual class i with a priority parameter γ_i can reach γ_i/e , where $0 \leq \gamma_i \leq 1$, regardless of the arrival rates of other classes. We have not been able to simulate these conditions because, when the total arrival rate exceeds $1/e$, the buffer for traffic class two packets increases to infinity which is unmanageable on a computer. This stability for high-priority traffic even if the whole system is unstable is very important since under bursty traffic conditions, the high-priority packet waiting time will remain bounded for a great range of total arrival rates. Finally, we can observe that for the adaptation algorithm, for low arrival rates, the high-priority delay is slightly longer than for the normal fixed frame length algorithm. However, as the arrival rate increases, the difference disappears. Thus,

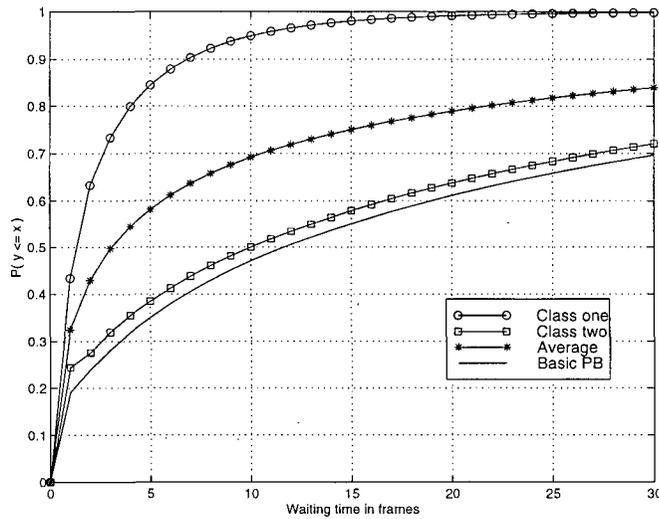


Figure 3.13: Cumulative density function of the waiting time for the SPBP system. $\lambda_1 = 0.15$ and $\lambda_2 = 0.20$ packets/slot. $\gamma_1 = 1$.

if the number of slots used for the contention slots is an important issue, the adaptation algorithm can be considered without being penalized by a significant degradation of the high-priority traffic class delay (for the low-priority traffic class the difference observed in Section 3.3.1, i.e., a one frame additional delay, remains).

The statistics presented so far are only average waiting time, but it is also important to see how the waiting times are distributed. Figures 3.13 to 3.15 show the *Cumulative Density Function* (CDF) of the three systems. Each CDF has been computed based on the waiting time in number of frames (for the slotted system, we can consider it as a framed system with 1 slot per frame). The presented results are for a total arrival rate of 0.35 packets/slot which is almost the saturation point for the non-priority pseudo-Bayesian algorithms. For the slotted system, we observe a great improvement for the high-priority traffic class while for the low-priority packets the performance is slightly better than the reference scheme. For example, for the SPBP system (Figure 3.13), 70%

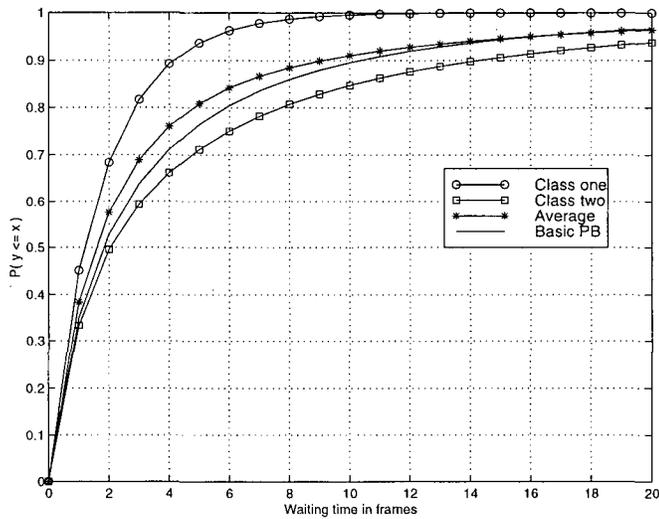


Figure 3.14: Cumulative density function of the waiting time for the FPBP system. $\lambda_1 = 0.15$ and $\lambda_2 = 0.20$ packets/slot. $\gamma_1 = 1$.

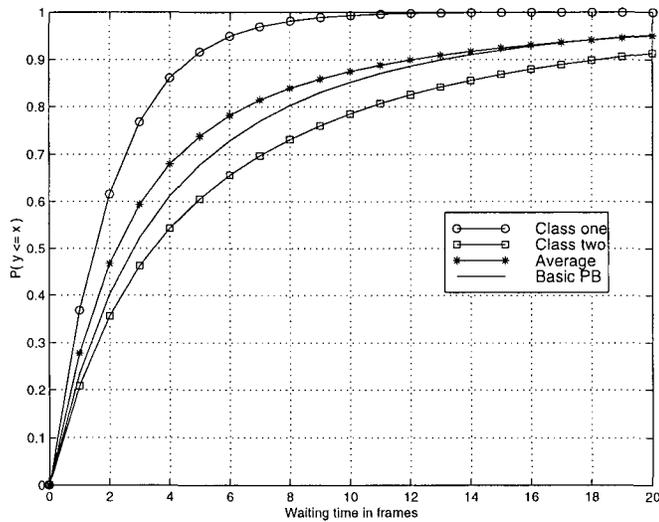


Figure 3.15: Cumulative density function of the waiting time for the FAPBP system. $\lambda_1 = 0.15$ and $\lambda_2 = 0.20$ packets/slot. $\gamma_1 = 1$.

of the high priority packets are transmitted less than 3 slots after their arrival, while for the reference algorithm, it is 30 slots. Furthermore, 90% of the high-priority packets are transmitted less than 7 slots after their arrival with the priority protocol while for the non-priority algorithm it is completely out of range.

For the framed system, the improvement is less spectacular but is still quite interesting. For example, for the FPBP system (Figure 3.14), we move from a point where 90% of the packets were transmitted less than 10 frames after their arrival in the reference system to a transmission under 4 frames for the priority algorithm, which represents an improvement of 60%. We observe a similar improvement of 60% for the FAPBP system (Figure 3.15) where the CDF at 90% goes from 13 frames to 4.5 frames.

3.4.2 Performance with More than Two Priority Classes

In this section, we have performed simulations to determine how the PSBP algorithm reacts when more than two priority classes contend for the channel. The system has been observed with three priority classes and we will see that we can easily predict the performance of the algorithm with a higher number of classes. The total arrival process is Poisson with intensity $\lambda_1 + \lambda_2 + \lambda_3$ and each packet has an independent probability $\lambda_i/(\lambda_1 + \lambda_2 + \lambda_3)$ of being part of traffic class i . The implemented prorating algorithm gives the following priority order: class one, class two and then class three. Thus even if two classes have the same priority parameter, a priority order will be given by the prorating algorithm. Other simulation conditions are similar to the ones presented in Section 3.4.1.

Figures 3.16 and 3.17 illustrate the impact of the class one priority parameter on the waiting time for the slotted system. The priority parameters of the two other classes were set for each simulated condition to $\gamma = (1 - \gamma_1)/2$. Thus no advantage is given to

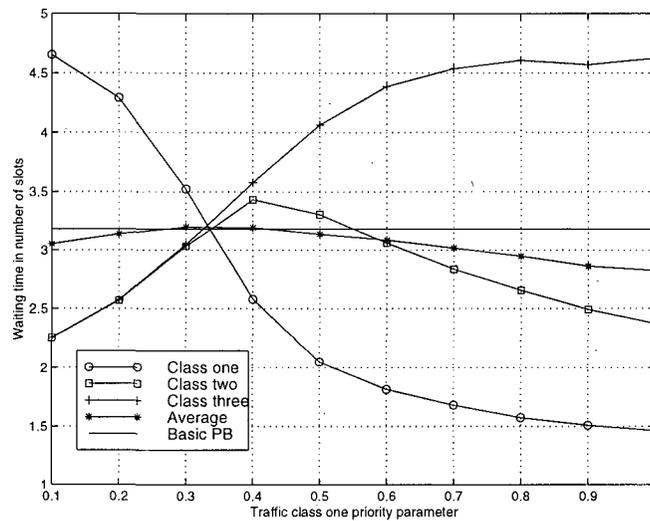


Figure 3.16: Waiting time as a function of the traffic class one priority parameter. $\lambda_1 = 0.08$, $\lambda_2 = 0.08$ and $\lambda_3 = 0.08$ packets/slot.

any of the low priority classes by the priority parameter. Only the prorating algorithm will implement the priority order among the two classes.

From these results many interesting conclusions can be drawn. First, we see that the algorithm is performing as expected with three priority classes. That is, when the priority parameter of traffic class one exceed its share of the total throughput, it has the lower waiting time of all traffic classes. We also observe that even if class two and three have the same priority parameter, class two traffic has a lower waiting time resulting from the priority order in the prorating algorithm. This confirms that the prorating algorithm is a key part of the performance of the priority system. We also remark that when the priority parameter of traffic class one exceeds its fair share, the waiting time stays low and stable for a great range of traffic intensity (waiting time between 1.5 and 3 slots observed at $\gamma_1 = 1$ for all conditions). Finally, we see that the optimum operating point is, like for the two priority classes case, at $\gamma_1 = 1$. At this point, the “average” waiting

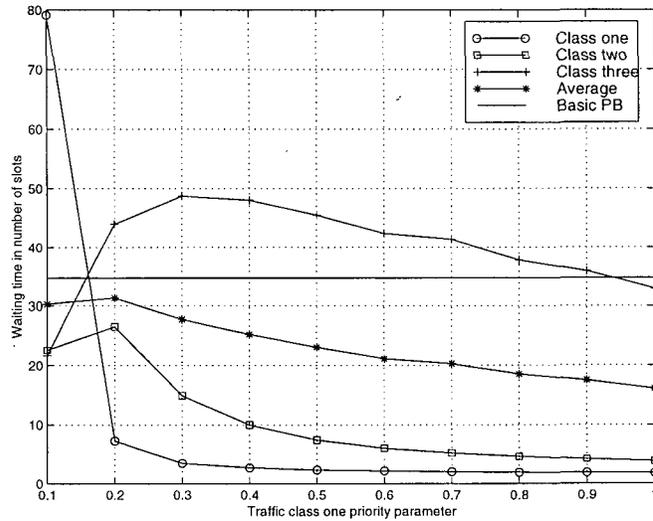


Figure 3.17: Waiting time as a function of the traffic class one priority parameter. $\lambda_1 = 0.05$, $\lambda_2 = 0.15$ and $\lambda_3 = 0.15$ packets/slot.

time is at its lowest value. We can observe at this optimum point a slight improvement for the high-priority and “average” waiting time over the two priority classes case. This confirms our hypothesis that the priority algorithm tends to split the contention channel in different sub-channels when we deviate from the fair share region. The number of collisions is therefore reduced and the waiting time decreases. However, we see that the low-priority waiting time is higher than it was before. This is normal since we have introduced a middle priority class between the high and low priority classes that takes some share of the channel from the low priority class without affecting the high priority traffic. From these results, we can easily generalize to the p priority classes case:

- The algorithm will perform as expected in the general case with p priority classes;
- The optimum operating point is at $\gamma_1 = 1$;
- The prorating algorithm will implement the priority order among classes i , $i \geq 2$;

- When the number of classes increases, the performance slightly improves.

The framed priority protocols implement the same algorithm that the slotted priority system uses. Therefore, the same conclusions can be drawn for the FPBP and FAPBP algorithms.

3.5 Simulation Results with Self-Similar Traffic

The pseudo-Bayesian algorithm (either with or without priority) is derived with the fundamental Poisson traffic assumption. However, it has been observed that real data traffic does not behave like a Poisson process. Instead, some authors demonstrated that many types of traffic like Ethernet traffic, wide area traffic and VBR video traffic exhibit long-range dependence and self-similar characteristics [54, 55, 56, 57]. In order to validate pseudo-Bayesian algorithms for an utilization in the “real world”, we must submit them to non-Poisson traffic to observe their robustness. We have thus ran simulations of the pseudo-Bayesian algorithms with and without priority with an asymptotically self-similar arrival process of data packets.

It has been shown that multiplexing constant-rate connections that have a Poisson arrival process and a heavy-tailed distributed connection lifetimes with infinite variance will result in an asymptotically self-similar data traffic [55]. We can show that the following probability density function satisfies the property of a heavy-tailed distribution with an infinite variance:

$$p[X = x] = 4/(x(x + 1)(x + 2)) \quad \text{for } x \geq 1 \quad (3.46)$$

For a discussion on the generation of self-similar traffic and the proof that equation (3.46) is a heavy-tailed distribution with infinite variance see Appendix A.

Even if a Poisson traffic process is bursty on a short period of time, if we observe this process on a large time scale all the bursts will disappear and the traffic will appear to be at a constant rate. This property allows us to obtain steady-state value of the statistics if we simulate the system on a relatively long period of time. On the other hand, self-similar traffic exhibits bursts of traffic on any time scale (this visual difference between a self-similar process and a Poisson process is well illustrated in [54]). This “burstiness” of a self-similar process makes it necessary that, even if the system is ergodic, we have to simulate it for an extremely long period of time in order to obtain steady-state statistic values. Instead, we have used a Monte-Carlo simulation approach to find the mean values of the waiting time with a confidence interval.

Each experimental conditions has been simulated 400 times with a length of one million slots each run. For each simulation, average waiting time statistics have been computed. In each sample experiment, the sample means can be assumed to be computed from a large number of iid means on observed subintervals of the total number of packets (in fact, since the arrival process is self-similar and exhibits some long-term correlation, each observations are not completely independent; however since the number of observations is large the independence assumption is good). Using the central limit theorem, we can thus assume that each sample mean is approximately Gaussian. According to [58], the $(1 - \alpha) \times 100\%$ confidence interval, for a particular sample of d observations $\mathbf{X} = (X_1, \dots, X_d)$, is then given by:

$$(M_d - z_{\alpha/2, d-1} V_d / \sqrt{d}, M_d + z_{\alpha/2, d-1} V_d / \sqrt{d}) \quad (3.47)$$

where:

$$M_d = \frac{1}{d} \sum_{j=1}^d X_j \quad \text{and} \quad V_d^2 = \frac{1}{d-1} \sum_{j=1}^d (X_j - M_d)^2 \quad (3.48)$$

and $z_{\alpha/2, d-1}$ is the value for which $\alpha = 2F_{d-1}(-z_{\alpha/2, d-1})$ where F_{d-1} is the Student's t-distribution CDF with $d - 1$ degrees of freedom.

In order to obtain comparable statistics between the non-priority and priority pseudo-Bayesian algorithms, for each simulation run, both algorithms have been simulated with the same traffic scenario. For the non-priority algorithm, even if the packet type does not influence the algorithm, the waiting time for a type of packet can significantly differ from the total average waiting time for a particular traffic scenario. We thus computed separately the average waiting time of each traffic class in the non-priority algorithm. This allows a better comparison between the two algorithms waiting time. For each traffic scenario, the average waiting time improvement for traffic class one packets (high priority packets) was compiled. Since, the average waiting time obtained for a simulation run for each algorithm is assumed to be Gaussian, then the difference will also be Gaussian and we can thus use the confidence interval given by equation (3.47).

To generate a traffic scenario with a traffic rate of λ_i (packets per slot) for priority class i , we produced a Poisson arrival process with intensity λ_i^c of connections of priority class i such that $\lambda_i^c E[X] = \lambda_i$ (where X is distributed as (3.46)). Then each new connection generates one packet of priority class i per slot for a random period of time (the connection lifetime) drawn from the probability distribution given by equation (3.46).

To evaluate how the algorithm is robust to data traffic different from Poisson traffic (in our case self-similar traffic), we have to simulate the system with conditions similar to the one that we will use if the traffic was Poisson. The algorithms are thus the same as they were with the Poisson traffic, the only change in the simulation is the arrival process. The arrival rate estimation is still made with a moving time-average of successful transmissions for each traffic class over a window period of 500 slots. Finally, the traffic class one priority parameter is set to its optimal value of one.

Table 3.1 presents the results obtained for average traffic class one waiting time for the slotted system while Table 3.2 gives the average waiting time obtained by considering both classes together. Each result is given with its 90% confidence interval. Results for

Table 3.1: Traffic class one waiting time in number of slots for slotted pseudo-Bayesian algorithms with self-similar traffic (90% confidence interval)

λ_1	λ_2	Poisson Traffic (Non-Priority)	Non-Priority Algorithm	Priority Algorithm	Improvement
0.05	0.05	0.961	16.9 (14.3–19.6)	13.5 (11.2–15.7)	3.46 (3.05–3.88)
0.05	0.10	1.40	25.3 (21.4–29.2)	14.9 (12.2–17.7)	10.3 (9.00–11.6)
0.05	0.15	2.17	35.7 (32.3–39.1)	14.8 (12.9–16.6)	20.9 (18.8–22.9)
0.05	0.20	3.61	54.1 (49.1–59.2)	16.1 (13.8–18.4)	38.0 (34.9–41.2)
0.05	0.25	7.54	102 (95.3–109)	15.9 (14.3–17.6)	85.9 (80.4–91.4)
0.10	0.10	2.17	33.2 (29.7–36.8)	18.7 (16.4–21.1)	14.5 (13.2–15.8)
0.10	0.15	3.61	57.4 (52.0–62.7)	21.7 (19.0–24.3)	35.7 (32.7–38.7)
0.10	0.20	7.54	102 (96.0–108)	21.7 (19.8–23.7)	80.5 (75.7–85.3)
0.15	0.15	7.54	106 (100–113)	28.9 (26.6–31.2)	77.4 (72.8–82.0)

the pseudo-Bayesian algorithm without priority with Poisson traffic are also presented to have a point of comparison. From these results, we easily see that there is a degradation of the waiting time for each protocol with self-similar traffic compared to the waiting time obtained with Poisson traffic. However, this is normal since self-similar traffic is a lot more bursty than Poisson traffic and thus some packets can experiment very long delays. Nevertheless, we observe that the priority scheme performs well with respect to the non-priority algorithm. We can see that with the priority scheme the waiting time for high priority packets is quite steady for any traffic conditions. It must be underlined that for every simulated traffic condition, the priority scheme provides an improvement for the class one waiting time for all traffic scenario. Finally, while with Poisson traffic we have observed an improvement of the average waiting time with the priority algorithm, with self-similar traffic there is a small degradation of the average waiting time. However, this degradation is overwhelmed by the great waiting time improvement given to the priority traffic with the priority algorithm.

Table 3.2: Average waiting time in number of slots for slotted pseudo-Bayesian algorithms with self-similar traffic (90% confidence interval)

λ_1	λ_2	Poisson Traffic (Non-Priority)	Non-Priority Algorithm	Priority Algorithm
0.05	0.05	0.961	16.2 (14.4–18.0)	16.9 (15.1–18.8)
0.05	0.10	1.40	25.9 (23.2–28.7)	27.5 (24.6–30.4)
0.05	0.15	2.17	39.1 (35.3–43.0)	41.9 (37.9–45.9)
0.05	0.20	3.61	55.4 (51.2–59.6)	61.0 (56.3–65.7)
0.05	0.25	7.54	106 (99.0–112)	121 (113–128)
0.10	0.10	2.17	33.4 (30.6–36.2)	36.4 (33.3–39.4)
0.10	0.15	3.61	57.0 (52.7–61.4)	64.4 (59.4–69.4)
0.10	0.20	7.54	102 (96.4–108)	122 (115–129)
0.15	0.15	7.54	106 (100–113)	130 (122–138)

We have then simulated the non-priority and priority framed pseudo-Bayesian algorithms with the same kind of traffic used for the slotted systems. The only difference is that a connection generates a packet per frame during its lifetime instead of one packet per slot as it was with the slotted system. We have used a frame length of 10 slots. Table 3.3 presents the results obtained for average traffic class one waiting time for the framed system while Table 3.4 gives the average waiting time obtained by considering both classes together. Each result is given with its 90% confidence interval. Results for the framed pseudo-Bayesian algorithm without priority with Poisson traffic are also presented to have a point of comparison. Although the framed pseudo-Bayesian algorithms present a performance degradation when they are used with self-similar traffic with respect to the results obtained with Poisson traffic, the performance degradation is less drastic than the one observed with their slotted counterpart. The priority scheme still provides a waiting time improvement for high priority traffic under all simulated conditions. Finally, in the framed environment, we see that the average waiting for the

Table 3.3: Traffic class one waiting time in number of frames for framed pseudo-Bayesian algorithms with self-similar traffic (90% confidence interval)

λ_1	λ_2	Poisson Traffic (Non-Priority)	Non-Priority Algorithm	Priority Algorithm	Improvement
0.05	0.15	1.35	1.68 (1.68–1.69)	1.53 (1.52–1.53)	0.16 (0.15–0.16)
0.05	0.20	1.55	2.59 (2.57–2.62)	1.81 (1.80–1.81)	0.79 (0.77–0.81)
0.05	0.25	1.98	6.48 (6.18–6.77)	2.16 (2.16–2.16)	4.32 (4.03–4.61)
0.05	0.30	4.74	34.0 (32.1–35.9)	2.60 (2.60–2.60)	31.4 (29.5–33.3)
0.10	0.10	1.35	1.68 (1.68–1.69)	1.53 (1.53–1.53)	0.15 (0.15–0.16)
0.10	0.15	1.55	2.58 (2.56–2.60)	1.81 (1.81–1.81)	0.77 (0.75–0.79)
0.10	0.20	1.98	6.16 (5.97–6.35)	2.16 (2.16–2.16)	4.00 (3.81–4.19)
0.10	0.25	4.74	33.9 (32.2–35.5)	2.59 (2.59–2.59)	31.3 (29.6–33.0)
0.15	0.15	1.98	6.11 (5.91–6.31)	2.22 (2.21–2.22)	3.89 (3.69–4.09)
0.15	0.20	4.74	33.4 (31.8–35.1)	2.64 (2.64–2.64)	30.8 (29.1–32.5)

Table 3.4: Average waiting time in number of frames for framed pseudo-Bayesian algorithms with self-similar traffic (90% confidence interval)

λ_1	λ_2	Poisson Traffic (Non-Priority)	Non-Priority Algorithm	Priority Algorithm
0.05	0.15	1.35	1.68 (1.68–1.69)	1.68 (1.68–1.69)
0.05	0.20	1.55	2.60 (2.58–2.62)	2.59 (2.57–2.62)
0.05	0.25	1.98	6.40 (6.18–6.62)	6.41 (6.18–6.64)
0.05	0.30	4.74	34.0 (32.6–35.9)	34.9 (32.9–36.8)
0.10	0.10	1.35	1.68 (1.68–1.69)	1.68 (1.67–1.68)
0.10	0.15	1.55	2.59 (2.57–2.61)	2.57 (2.55–2.59)
0.10	0.20	1.98	6.16 (5.98–6.35)	6.09 (5.90–6.28)
0.10	0.25	4.74	34.0 (32.3–35.7)	34.2 (32.4–36.0)
0.15	0.15	1.98	6.13 (5.92–6.33)	5.95 (5.76–6.14)
0.15	0.20	4.74	33.6 (31.9–35.3)	32.2 (30.5–33.9)

non-priority and priority algorithms are similar.

We can conclude that the pseudo-Bayesian algorithms have a better reaction to self-similar traffic in the framed environment than in the slotted one. This can be explained by the fact that the framed algorithm will try to spread the burst of data over the entire frame by its random slot selection process. Furthermore, we have demonstrated that the priority algorithm is robust to the type traffic that exhibits the most burstiness (i.e. self-similar traffic). It is thus reasonable to predict that it will be robust to any other kinds of traffic.

Chapter 4

Integrated Resource Allocation

Algorithm for DR-TDMA

In Section 2.5, we have proposed a novel DR-TDMA MAC protocol for a wireless ATM network. One of the key feature of the DR-TDMA protocol is the framed Pseudo-Bayesian priority Aloha algorithm introduced in Section 3.3.2 that minimizes the contention delay and provides priority services. The other main feature of the DR-TDMA WATM MAC protocol is the integrated resource allocation algorithm presented in this chapter. Once connections have transmitted their bandwidth requests to the base station, either in control packets or piggybacked to WATM data cells, the DR-TDMA scheduling algorithm allocates the TDMA slots to the connections according to their required bandwidth and QoS while maximizing the wireless channel utilization. The proposed resource allocation algorithm integrates CBR, VBR and ABR traffic while maintaining their respective QoS. It also uses the FPBP algorithm to manage the control slots access and to dynamically adjust the uplink control period length. Furthermore, we introduce a novel rate-based VBR allocation algorithm featuring a cell control algorithm to define the flow conformance to the VBR traffic contract. Cells that conform with the traffic parameters receive a guaranteed service while the non-conforming cells take advantage of unused resources with a best effort service strategy. Simulation results illustrating the performance of the DR-TDMA MAC protocol with different traffic scenarios are



Figure 4.1: Bandwidth allocation problem

presented at the end of the chapter.

4.1 Bandwidth Allocation for Wireless ATM

The main objective of a wireless ATM bandwidth allocation algorithm is to efficiently exploit statistical multiplexing while maintaining the negotiated quality of service of each admitted connection in the network. In wired ATM, to achieve statistical multiplexing, the switch scheduler has a direct access to stream buffers. However, in wireless ATM the stream buffers are distributed within the mobile terminals and the scheduler does not have a direct access to the buffer state information. Furthermore, the wireless ATM MAC scheduler must deal with the difficulties imposed by the TDMA frame structure.

Figure 4.1 illustrates the bandwidth allocation problem for uplink and downlink slots in a frame $t + 1$. For uplink allocation, a mobile must observe its dynamic parameters in frame $t - 1$, transmit them to the base station in frame t and receive the slot allocation in frame $t + 1$. Any delay in transmitting information from the mobile to the base station scheduler will degrade the allocation algorithm performance. On the other hand, for downlink allocation, the base station has a direct access to the downlink stream buffers. From the observation made in frame t , it can send downlink slots transmission announcement at the beginning of frame $t + 1$. However, since the allocation (both for uplink and downlink) in a frame can only be decided at the beginning of the MAC frame (for reservation based protocol), any unutilized slots cannot be reallocated based on the

instantaneous requirements of the other streams.

The bandwidth allocation problem can be divided into two parts: the scheduling information and the scheduling algorithm. In the following subsections, we will try to describe these two problems and the solutions that have been proposed in the literature.

4.1.1 Scheduling Information

A scheduling algorithm must decide, according to the requirements and characteristics of each connection, how to allocate the available bandwidth based on its knowledge of the system state. Thus, two types of information are needed by the base station scheduler: characteristics of admitted connections and dynamic state of the system.

When a new connection is established, it transmits its characteristics to the base station. These consist of the type of data, quality of service requirements and physical limits. More precisely, the following list illustrates some connection characteristics that can be useful to a scheduling algorithm:

- Data type:
 - CBR, VBR or ABR;
 - Delay sensitive data or not;
 - Real time traffic or not;
 - Priority;

- Cell rate:
 - Minimum cell rate;
 - Sustained cell rate;
 - Source peak cell rate;

- Equivalent bandwidth (for VBR traffic);
- Leaky bucket parameters;
- Delay:
 - Maximum cell delay;
 - Cell delay variation (jitter);
- Wireless cell loss rate;
- Maximum cell rate supported by the mobile radio transmitter [13].

In order to determine slot allocation, the base station scheduler must know the actual network state. The capacity requirements of each connection, buffers state and radio link measurements are some elements of the network state. The following list enumerates dynamic information that can be used by the scheduling algorithm to determine the network state:

- Required bandwidth (can be explicit requirement or binary feedback: more or less than last allocation);
- Arrival of a new message (with length of the message);
- Instantaneous cell arrival rate;
- Number of arrivals in the last frame;
- Buffer state:
 - Number of queued cells;
 - Increase or decrease of buffer length;

- Discrete buffer state (empty, relatively empty, full, overflow, ...);
- Residual lifetime of most critical queued cell;
- Mean residual lifetime of n most critical queued cell;

- Radio link bit error rate;
- Retransmission of corrupted wireless cells;

- Power allocation [13];

- Allocation-utilization ratio (fraction of allocated slots which are actually used);

- Time of the last arrival.

These information can be measured directly by the base station where possible, or transmitted from the mobile to the base station either in uplink control slots or piggybacked to information cells. For the cases where the dynamic state parameters of the system must be transmitted from the mobile to the base station, the amount of information, the encoding of the information and the frequency at which the information is transmitted must be selected to reduce the channel traffic and the overhead while allowing the base station to maintain an accurate picture of the network state.

4.1.2 Review of Proposed Scheduling Algorithms

Scheduling algorithms differ depending on which type of traffic it must schedule. CBR, VBR and ABR traffic are treated in different ways, although for a multimedia environment, the bandwidth scheduler must integrate these traffic types since the radio resource is shared among them. In the literature, we have found only two schemes that integrate CBR, VBR and ABR traffic in the channel allocation. Most of the other proposals only consider one or two types of traffic.

CBR Allocation

CBR traffic is the easiest type of traffic to schedule. Since the fixed stream arrival rate is declared at the connection admission, the base station just have to allocate the required number of slots in each frame. The scheduler will try to keep the position of the assigned slot within a frame relatively static to facilitate the operation of low complexity terminals. The most obvious allocation scheme is the one used in PRMA where the same slot in subsequent frames is allocated to a mobile during the entire length of the connection [21].

However, in wireless ATM the bit rate is not identical for each connection and might not coincide with the frame length (i.e. the number of required cells per frame is not an integer). In [5], an algorithm is proposed to solve the latter problem. Periodically, an additional reservation is added in a frame in order to provide the required bit rate over multiple frames. A similar approach is used in [37] where an embedded frame scheme is used.

In C-PRMA [24], the scheduling algorithm is based on the maximum delay before the next packet in the connection stream will be discarded. The next transmission is thus scheduled before the expiration time of the packet; if it is not possible, the packet is discarded. Finally, in [59] it is proposed to use the time of the last arrival to predict the time of the next arrival, since the inter-arrival time is fixed in CBR.

Voice connections with a voice activity detector can also be considered as CBR connections. The only difference is that the bit rate is constant over a talkspurt, thus the above algorithms can be used on a talkspurt basis. To maintain the quality of service, a mechanism allowing a mobile to rapidly inform the base station of the beginning of a talkspurt is required. The mobile also informs the scheduler of the end of a talkspurt with a piggybacked bit for example.

ABR Allocation

For ABR traffic not many algorithms are proposed. Mainly the proposed schemes are similar with what is actually used in wired networks for data traffic. The algorithm must be efficient and also fair between each mobile (i.e. a mobile with a lot of data must not prevent other users to have access to the channel). When a mobile receives new cells it informs the base station of the number of arrivals using uplink control slots or it can piggyback with each cell the number of ABR packets queued at the mobile. Based on this information, the scheduler can implement one of the following algorithms to allocate slots:

- *First In First Out* (FIFO);
- Fair queuing;
- Burst service;
- Time-of-expiration based queue service [1].

VBR Allocation

For VBR allocation, two approaches are proposed: algorithms specifically designed for VBR traffic or a combination of CBR fixed periodic assignment with ABR dynamic allocation. The latter approach is proposed in [1, 27], however no concrete algorithm is described.

In [60], the author proposes to use a prediction field in the cell header for explicit scheduling of the next cell. For layered VBR sources (e.g. video), the author in [32] proposes to use a VBR allocation algorithm providing both guaranteed and best effort services. It has been shown that, for coder generating two-layer priority coding, as long as the baseband cells are received, the enhancement-layer cells can be discarded

without significantly reducing the picture quality. Thus, only the baseband layer needs a guaranteed service and the enhancement-layer cells can receive a lower QoS. In addition, retransmissions are only needed for baseband packets, further reducing the traffic on the channel. Although the cell arrival prediction and VBR service layering are interesting concepts, no concrete algorithm are presented in the papers.

For the NEC WATM MAC protocol, an adaptive wireless slot allocation scheme that tracks the short and long term rate variations of VBR virtual circuits is proposed in [29]. The slot allocation estimation uses instantaneous traffic rate, buffer state and allocation-utilization information. The estimation philosophy is motivated by the fact that the rate of a typical video stream is constant over a video frame which is usually longer than the MAC frame length. While for downlink connections the base scheduler has direct access to input rate and buffer state such that the estimation can be performed in a straightforward way, for uplink connections the estimation algorithm is determined by the adopted control strategy. Two uplink estimation control schemes are proposed in the paper. In the first one, estimation for a connection is performed within the source mobile terminal using the input rate and buffer state information. The estimated allocation requirement is then sent to the base scheduler by using an *out-of-band* control message. The other approach uses an *in-band* signaling to transmit a compressed version of the buffer state information from the mobile to the base station. Allocation estimation is then centrally made in the base station scheduler using allocation-utilization ratio and compressed buffer state information. Simulation results show that the maximum throughput that the proposed protocol can sustain while maintaining reasonable QoS remains low (in the range of 60 to 70%). Furthermore, when the number of VBR connections in the system is low, they have a problem of over-allocation (i.e. too many slots are allocated for VBR traffic and are therefore not available to lower priority traffic like ABR).

Integrated Traffic Allocation Algorithm

In a mixed traffic environment, each traffic category has its own traffic properties and service requirements. Thus, in the slot assignments different priority must be assigned for each connection in order to maintain the required QoS. Generally speaking, CBR traffic has priority over VBR connections and those have priority over ABR data.

For the DSAMA protocol [33], an integrated scheduling algorithm is presented. The types of traffic considered are: voice traffic, VBR (video) traffic and data. VBR sources are characterized by their equivalent bandwidth (an ON/OFF source model is used), their buffer status (empty or full) and their instantaneous slots requirements. The scheduler will try to allocate slots to each connection in this order: voice connections, buffer full video connections, buffer empty video connections who need less than their equivalent bandwidth, buffer empty video connections that need more than their equivalent bandwidth, and data connections. Slots will be release from data connections and buffer empty video connections if there is not enough slots for voice and buffer full video connections.

A similar approach is presented for the PRMA/DA protocol [30]. The base station scheduler requires the following parameters: the average traffic rate , the peak rate and the number of requested slots by each connection. First, every CBR connection is assigned slots according to its declared bit rate at admission. Second, a number of slots equal to the sum of the average traffic rate of VBR connections is assigned to VBR traffic. If the leftover is not totally used by the ABR connections, the remainder is reallocated to VBR connections. The allocated slots are then distributed, in each group, according to the average traffic rate of individual connections. Surplus slots from connections requiring less bandwidth will be distributed to connections requesting more slots. An algorithm to dynamically determine the required number of control slots in each frame is also presented.

These two integrated scheduling algorithms present some pitfalls. First, both of them need the exact number of required slots to perform their allocation. However, transmitting the required number of slots will require a lot of overhead (each cell must include this field) or a lot of traffic in the uplink control slots. Furthermore, how these requirements are calculated is not presented and, if no estimation is made, this number will represent the needs of two frames before (as explained at the beginning of Section 4.1).

4.2 Multimedia Source Characteristics

The CBR, VBR and ABR wireless ATM services can be categorized into two basic traffic type: *Real Time Traffic* (RTT) and *Non-Real Time Traffic* (NRTT). CBR service (low complexity voice terminal) and VBR service (voice terminal with voice activity detector, compressed video) are RTT type sources. If a RTT packet is not delivered to its destination within its *Maximum Transfer Delay* (MTD), it would be dropped. In general, the RTT performance is measured by the cell dropping or loss probability and the cell delay. Cell jitter performance is not considered since it can be corrected by using a leaky bucket at the interface between the wireless hop and the wired network. On the other hand NRTT (ABR service) is more tolerant to delay but has stringent requirements for cell loss probability. The NRTT performance is described by its cell delay.

In this section, we present the models used to describe the wireless ATM sources that we have considered in this thesis: a voice source, a data source and a VBR source. The allocation for CBR sources is trivial: periodic slots are allocated to a connection according to its declared bit rate during its lifetime. Although an allocation algorithm for CBR traffic is described in Section 4.3.5, we have not considered the effect of CBR sources in our research and no source model is presented. The data source model is used to represent ABR traffic. However, we did not considered the ABR congestion control

mechanism [61, 62, 63, 64] since to evaluate it correctly, we have to integrate the wireless link in a complete network which is not our goal in this research.

4.2.1 Voice Source Model

The simplest model for a voice source is to represent it as a CBR source that transmits packets at the same rate for the entire duration of the connection. However, a voice source generates a signal that follows a pattern of talkspurt and silent gaps and a speech activity detector can be used to detect this pattern. Data transmission can thus be stopped during periods of voice inactivity to reduce the traffic and increase the statistical multiplexing. Therefore, a voice source can be described by an ON/OFF model. An ON/OFF source alternates between two states: the ON state where the source generates packets at rate R_v and the OFF state where no packets are generated. ON and OFF state durations are modeled by exponential distributions with means t_b^{vo} and t_i^{vo} , respectively. This model is similar to the “slow” speech activity detector model described in [65] and we have selected the same parameter values for t_b^{vo} and t_i^{vo} .

A voice source is also characterized by the encoder bit rate R_v and the packetization delay to assemble a packet payload T_v . In our case we have a payload of 48 bytes, we thus have $T_v = 384/R_v$. The voice activity ratio of a voice source is defined as $t_b^{vo}/(t_b^{vo} + t_i^{vo})$ and represents the average number of packet transmitted by the source per period T_v . Our voice source model has a one packet buffer. That is, when a new voice packet is generated, if the previous packet has not been transmitted, the buffered packet is discarded. The MTD is thus equal to the packetization delay T_v . Finally, voice request packets are assigned a high priority for the contention protocol. Table 4.1 summarizes the voice source parameters.

Table 4.1: Voice source model parameters

t_b^{vo}	1.00 s
t_i^{vo}	1.35 s
Activity ratio	0.426
R_v	24 Kbps
T_v	16 ms
MTD	16 ms
Priority	high

4.2.2 Data Source Model

Data sources are represented by a model where groups of packets arrive in the buffer at a certain rate. This model is in accordance with the AAL5 layer that will be used for ABR and *Unspecified Bit Rate* (UBR) traffic. When AAL5 receives a packet from an upper layer, it segments the packet in ATM cell of 48 bytes (or wireless ATM cells in our case). It is thus reasonable to assume that these cells arrive in the wireless ATM MAC data buffer at approximately the same time since the packetization rate will be much faster than the channel rate.

The interarrival time between two groups of packets is exponentially distributed with mean t_d . The number of packets in a batch arrival is given by $b = \lceil x \rceil$ where x is gamma distributed with parameter β and θ . That is, the random variable x is the sum of β independent, exponentially distributed random variables, each with mean $1/\beta\theta$. With the use of a random number generator, we can find that b has a mean μ_b approximately given by $1/\theta + 0.5$ and a mode value of $\frac{\beta-1}{\beta\theta} + 0.5$. We have selected this model instead of the usual geometric model because the later one has a mode value of one which is not realistic for ABR traffic.

Even if data are non-real time traffic, we also assign a maximum transfer delay to the packets. This delay is chosen long enough such that it would not cause any loss

under normal operation. But when the load is too heavy, this allows the old packets to be discarded. In fact, this is similar to have a finite buffer length. Finally, data request packets are assigned a low priority for the contention protocol.

4.2.3 VBR Source Model

A VBR source generates a traffic flow at a rate which varies with time. If the transmission rate takes continuous values and varies in an uncontrolled manner, it would be complex and require a heavy control traffic from the VBR source to the base station in order to implement an efficient scheduling algorithm. This is due to the fact that the resource allocation algorithm needs to know what is the buffer state in the VBR source in order to efficiently schedule slots while respecting the VBR connections QoS. We have therefore decided to use a discrete rates VBR model. In this model, the source traffic flow, after cell c arrival, has an instantaneous cell rate $l^c R_b$ where $l^c = 0, 1, \dots$ is the transmission rate level and R_b is the basic source cell rate. When cell c arrives, the cell arrival rate changes to the value $l^{c+1} R_b$ (n.b. $l^{c+1} R_b$ can be equal to $l^c R_b$) and packet $c + 1$ will arrive $1/(l^{c+1} R_b)$ seconds after cell c arrival. This model is conforming with real VBR sources. For example, a VBR video coder can be forced to transmit cells using a limited number of transmission rates [66, 67]. A VBR traffic flow can also be rate-controlled using algorithms similar to the ones presented in [68].

The rate-controlled VBR arrival traffic can be modeled, similarly to what was proposed in [69], by a superposition of S ON/OFF sources. Each ON/OFF source s ($s = 1, \dots, S$) alternates between the ON and OFF states. ON and OFF state durations are modeled by exponential distributions with means t_b^{vbr} and t_i^{vbr} , respectively. This mathematical model can not represent every types of VBR connections but is good enough to show the protocol efficiency and properties. The parameter l^c is equal to the

number of sources simultaneously in the ON state at discrete time c .

Suppose that there are l^c ($l^c = 1, \dots, S$) ON sources after the transmission of packet c , then each of the l^c ON sources ends its ON state before the transmission of packet $c + 1$ with probability

$$p_{\text{ON-OFF}} = 1 - e^{-1/(l^c R_b t_b^{vbr})} \quad (4.1)$$

Similarly, each of the $S - l^c$ OFF sources ends its OFF state before the transmission of packet $c + 1$ with probability

$$p_{\text{OFF-ON}} = 1 - e^{-1/(l^c R_b t_i^{vbr})} \quad (4.2)$$

Let $S_{\text{ON-OFF}}$ be the number of ON sources that ended their ON state before the transmission of packet $c + 1$ and $S_{\text{OFF-ON}}$ the number of OFF sources that ended their OFF state before the transmission of packet $c + 1$, then

$$l^{c+1} = l^c - S_{\text{ON-OFF}} + S_{\text{OFF-ON}} \quad (4.3)$$

If $l^c = 0$, then after a time

$$\gamma = \min_{s=1, \dots, S} (T_s) \quad (4.4)$$

where T_s is the OFF state length of source s (exponentially distributed with mean t_i^{vbr}), l^c changes to 1. Packet $c + 1$ is then transmitted after a time $1/R_b$ and the state transition probabilities are computed as described above.

The VBR arrival flow is thus characterized by the following parameters of the superposition of ON/OFF sources model:

- R_b : the basic cell rate of one ON/OFF source;
- S : the number of ON/OFF sources;

- t_b^{ubr} : the mean ON state length of one ON/OFF source;
- t_i^{ubr} : the mean OFF state length of one ON/OFF source.

With this VBR source model, the average cell rate is:

$$R_c = \frac{t_b^{ubr}}{t_b^{ubr} + t_i^{ubr}} S R_b \quad \text{cells/second} \quad (4.5)$$

and the average connection bit rate is $384R_c$ bits/second if we considered an ATM payload of 48 bytes per cell. Finally, the maximum bit rate is $R_{max} = 384SR_b$ bits/second.

When a VBR connection is admitted in the wireless ATM network, it is required to specify the traffic parameter of its transmission flow. Cells that conform to the traffic contract of their VBR connection receive a high quality of service (guaranteed service) while non-conforming cells receive a low quality of service (best-effort service). Non-conforming cells are not discarded to allow them to take advantage of unused resources in the network. Figure 4.2 shows the cell control algorithm that allows packets that conform with the declared guaranteed VBR traffic parameter to be placed in the guaranteed buffer while non-conforming cells are put in the best-effort buffer. Cells are queued in the guaranteed and best effort buffer until they receive a slot allocation for transmission over the wireless channel. In the cell control algorithm, guaranteed tokens arrive at a constant rate of gR_b and are queued in the guaranteed token pool. The token pool can store a limit of W tokens. If a guaranteed token arrives when there are W tokens in the token pool, the arriving token is lost.

When a packet arrives from the VBR rate-controlled source, the cell control algorithm determines if the packet conforms with the guaranteed traffic parameter. An arriving packet is considered to comply with the guaranteed traffic parameter if, when the packet arrives from the source, there is a guaranteed token in the guaranteed token pool. If it is the case, a token is removed from the guaranteed token pool and the packet is placed

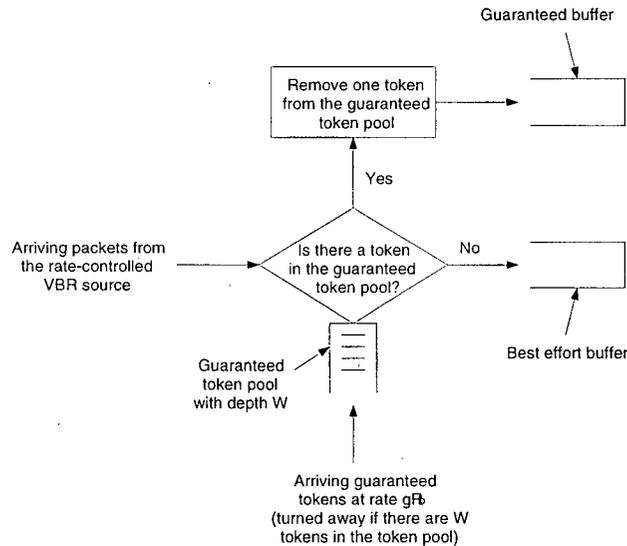


Figure 4.2: Cell Control Algorithm

in the guaranteed buffer and will receive a high QoS. Otherwise, if the guaranteed token pool is empty, the packet is put in the best effort buffer and will receive a low QoS.

The cell control algorithm that we use is quite similar to the virtual scheduling algorithm proposed by the ATM Forum [61]. The only difference is that in our model, the next token arrival time is determined by the last token arrival time while in the ATM Forum proposal the packet arrival time is also considered. This modification facilitates the implementation of the prediction algorithm presented in Section 4.3.3 while barely affecting the traffic characteristics.

For the DR-TDMA WATM MAC protocol, a VBR connection is described by the following traffic parameters:

- R_b : the source basic cell rate;
- g : the parameter characterizing the sustained guaranteed cell rate gR_b ;
- W : the guaranteed token pool depth;

- *Peak Cell Rate* (PCR): maximum guaranteed cell rate;
- MTD: the maximum cell transfer delay from the best-effort or guaranteed buffer to the base station.

The connection PCR is determined by the rate-controlling algorithm, that is $\text{PCR} = n_{max}R_b$ where n_{max} is the maximum l^c that the rate-controlling algorithm allows. The cell control algorithm then determines the *Maximum Burst Size* (MBS) that may be transmitted at PCR and still be in conformance with the traffic contract. The MBS in number of cells is given by:

$$\text{MBS} = \left\lceil 1 + \frac{W \times \text{PCR}}{\text{PCR} - gR_b} \right\rceil \quad (4.6)$$

The token pool depth W can also be related to the *Burst Tolerance* (BT)[61] with the following relation:

$$\text{BT} = \frac{W}{gR_b} \quad (4.7)$$

If the VBR rate-controlled arrival flow (either from a video coder or from a rate-controlling algorithm) conforms to the guaranteed traffic parameter, the connection will receive a high QoS. Otherwise, only cells that conform to the guaranteed contract receive high QoS and extra cells receive a low QoS. When a queued packet exceeds its MTD, it is removed from the buffer. We have assumed that the guaranteed and best-effort queues have an infinite length. However, since packets are discarded when they exceed their MTD, overflow will be avoid if the queues' length are greater than $\text{MTD} \times \text{PCR}$. It can be noted that a voice source with a voice activity detector can be treated as a VBR connection with one ON-OFF source, $g = 1$ and $W = 1$.

4.3 DR-TDMA Allocation Algorithms

In this work we only considered the allocation for uplink connections since downlink allocation can be done like in a usual ATM switch because the base station has a direct access to the downlink connection buffers (the only slight difference is that allocation decisions in the wireless scheduler are made on a frame basis). However, this is not the case for uplink connections where the buffers are distributed among the mobiles and the multiplexer only has an indirect access to the buffer status information through piggy-backing, control packets and feedback. Therefore, downlink slot allocation is a simple problem compared to uplink allocation. Distribution of the number of slots between the uplink and downlink periods is simply done by considering the instantaneous request of the connections.

For our DR-TDMA MAC protocol architecture, the scheduler decides, based on its current knowledge of the network state and connections traffic parameters, how to assign slots to each connection and the number of slots to allocate for control traffic. In the following subsections, we describe the allocation algorithms for voice, data, VBR and CBR traffic and for contention traffic. Then the integration of these algorithms is described.

4.3.1 Slot Allocation Algorithm for Voice Traffic

The slot allocation algorithm for voice traffic uses a *Time-To-Expiry* (TTE) approach where the connection for which the voice packet will expire and be discarded first receive the slot allocation. When a new voice call is established, the mobile sends a request packet in the contention period using the FPBP protocol with high priority until success. This control packet contains the voice connection encoder bit rate from which the cell interarrival time can be computed in the base station. At the beginning of a new talkspurt, the mobile also sends a high priority control packet in the contention period

until success or the talkspurt ends. The control packet contains the time of arrival of the last voice packet. A connection is considered as active when the base station receives a control packet indicating the beginning of a talkspurt.

For each active connection, the base station records the time of the last voice packet arrival. The scheduler is thus able to find the waiting time experienced by a packet and when it will be lost. It can also predict the arrival time of the next packet for each active voice connection. When the voice activity detector detects the end of a talkspurt it will set a bit in the last talkspurt packet to indicate to the scheduler the beginning of a silent gap. The scheduler then removes the connection from the active set after the reception of the last talkspurt packet. If this packet is lost, a control packet is sent to the base station, either in contention or in the next slot allocated by the base station to this voice connection.

For each active connection, the base station keeps the time of the last packet arrival if it has not yet been transmitted or, otherwise, the time of the next arrival. Active voice connections are sorted in a list in increasing order of TTE where $TTE = MTD - (\text{Time} - \text{Arrival Time})$. Assume that A slots are available for voice transmissions. The scheduler then allocates slots to the connections that have waiting packets in order of increasing TTE. The allocation algorithm can be summarized as follows (in the algorithm the TTE always refers to the TTE of the first connection in the list):

```
while ( $A > 0$  and  $TTE \leq MTD$ )  
  /* TTE  $\leq$  MTD ensures that the packet has arrived in the mobile */  
  if ( $TTE \geq 0$ )  
    Allocate a slot to the first connection in the list  
     $A = A - 1$   
    if (This packet is the last of the talkspurt)
```

```
    Remove the connection from the active list
else
    Enter this connection in the list with the TTE corresponding
    to the time of the next arrival
else
    Packet loss
    Enter this connection in the list with the TTE corresponding
    to the time of the next arrival
```

4.3.2 Slot Allocation Algorithm for Data Traffic

For the data traffic slot allocation algorithm, the base station keeps the buffer length status of active connections (an active data connection is defined as a connection with a non-zero buffer length) and allocates slots to the connections using a fair bandwidth allocation algorithm. When a batch of packets arrives at a mobile during frame t , the mobile MAC controller sends a piggybacked request or a control packet depending on the mobile's queue state at the beginning of frame $t + 1$. This gated mode of operation is used because we assume that the wireless packets are formed at the beginning of the frame and can not be modified just before transmission to piggyback the new information. This is a realistic assumption since we do not know the processing time that this modification will required and it is also a worst case assumption. If the queue is empty at the beginning of frame $t + 1$, the mobile sends a low priority control request packet in the following contention periods using the FPBP protocol until success. Otherwise, the request will be piggybacked to the next information packet transmitted to the base station. The request contains the number of packets in the new batch arrival. This allows the base station to

keep the exact buffer length of each active connection. Since slot allocation is announced at the beginning of the frame, allocation for this new request can only start during the next frame. Thus, for a batch arrived in frame t and for which the request has been successfully received in frame $t + 1$, the slot allocation for these packets can only begin in frame $t + 2$. There is thus a minimum waiting time of $2T$, where T is the frame length, for data packets transmission.

Based on the current buffer length status of all active connections, the base station scheduler allocates the A available slots to connections according to the fair bandwidth allocation algorithm. The algorithm, adapted to the slotted environment (i.e. we must assign an integer number of slots instead of a continuous bandwidth), can be described as follows:

```

if (The sum of individual demands  $\leq A$ )
    Allocate requested demand to all connections
    Update connections buffer length
else
    while ( $A > 0$ )
        /* C is the number of data active connections */
        Fair= $\lfloor A/C \rfloor$ 
        if (Fair=0)
            Fair=1
        for (All active connections)
            if ( $A=0$ )
                Stop
            if (Connection buffer length  $\leq$  Fair)
                Allocate buffer length

```

```

    Update buffer length
    Remove connection from active set and  $C = C - 1$ 
     $A = A - \text{Buffer length}$ 
else
    Allocate Fair
    Update buffer length
     $A = A - \text{Fair}$ 

```

For example, assume that five connections have the following buffer length: [2,1,4,3,6] and that the number of available slots is 11. During the first allocation round, the fair share is 2 slots and connection 2 receives 1 slot and connections 1, 3, 4 and 5 receive 2 slots. There is still 2 slots available and the active connection set is now made of connections 3, 4 and 5. For the next allocation round the fair share is 1 and connections 3 and 4 receives 1 slot while connection 5 receives no slot. The allocation algorithm is then finished and the active connection set is composed of connections 3 and 5.

When a mobile receives its slot allocation, it can serve packets that are queued in its buffer in the order that it prefers. We have decided in this thesis to implement a FIFO queuing discipline for mobile's buffer. Finally, at the beginning of each frame, the base station scheduler will update the buffer length status of connections for which it has received a request in the last frame (either a control packet or a piggybacked request) and if the connection was not in the active connection set it will be added to the list.

4.3.3 Slot Allocation Algorithm for VBR Traffic

The rate-based VBR allocation algorithm maintains in the base station a virtual status of each admitted VBR connection queue (number of packets and time of arrival) and allocates slots according to the specified QoS and the VBR connections' virtual queue

in the base station. The current cell arrival rate of the VBR source is transmitted from the mobile to the base station piggybacked to data packets or in control packets and is used to predict the guaranteed and best effort buffer status of the connection. Our allocation strategy provides both guaranteed and best effort services for traffic parameters conforming and non-conforming cells, respectively. Guaranteed packets are served using a TTE approach while best effort cells are scheduled according to the fair bandwidth allocation algorithm.

When a new VBR connection is established, the mobile sends a high priority request packet in the contention period using the FPBP protocol. This packet contains the VBR traffic parameters, that is the source basic rate R_b , the guaranteed parameter g , the token pool depth W , the PCR and the MTD. These parameters can be re-negotiated during the course of the connection. The request packet also contains the initial cell rate, the first cell arrival time, the first guaranteed token arrival time and the initial number of token in the guaranteed token pool.

For each active VBR connection, the base station maintains the following status information:

- current cell rate;
- time of the next packet arrival;
- virtual queue of best-effort packets to transmit;
- virtual queue of guaranteed packets to transmit;
- time of the next guaranteed token arrival;
- number of tokens in the guaranteed token pool.

This status is predicted in the base station using information sent from the mobile to the base station piggybacked to data packet or in control packet transmitted in the uplink control period using the FPBP protocol.

Each arriving packet is assigned a sequence number that is required for the data link layer protocol and it will also be used by the MAC protocol to reorder the packets. A MAC field is used to indicate the packet type (i.e. best-effort or guaranteed) and the source rate-level change after the packet left the source (i.e. for packet c it is equal to $l^{c+1} - l^c$). The overhead required for the allocation algorithm in data packets is thus relatively small. One bit can indicate the packet type while the rate change can be coded in a small number of bits. For example, if we limit the variations of the source parameter l^c to an absolute value smaller than 8, then 4 bits are enough for this MAC field.

A control packet is transmitted to the base station during a connection when one of the following situations occurs:

- the connection wants to renegotiate its parameters;
- the cell rate was equal to zero and packet arrival resumes;
- a packet with a non-zero rate change field is dropped because it has exceeded the MTD.

For the first case, the control packet is similar to the initial connection request packet. For the two other cases, for each queued or dropped packet for which there was a non-zero rate change MAC field, the sequence number, the packet type and the rate change is included in the request packet and the corresponding MAC field in the queued packet is set to zero. Furthermore, for the cells where the preceding cell rate was zero, the packet arrival time is included. While the amount of information in request packets may seem

large, we should remember that the complete packet payload is used to transmit the information. And, even if there is not enough place in the request packet to store the complete information, it is the information about dropped and old packets that is most useful to the base station to retrack the actual status of the mobile's queues.

These information allow the base station to maintain a virtual status of the VBR source in the mobile. The rate change information piggybacked to data packets allows the base station to know the source cell arrival rate after the packet arrival in the mobile. The base station can therefore predicts when the subsequent packet arrivals occurred and future arrivals time. Furthermore, the scheduler can determine if the cells are guaranteed or best-effort packets. If a rate change occurs for one of the predicted packets, then subsequent prediction are erroneous. However, when the base station receives the packet with the rate change, a new prediction is done with the new received information and virtual buffer status are correctly updated. When a packet with a non-zero rate change field is dropped or cell arrival rate is equal to zero, the base station can not correctly predict cell arrival and buffer status, this is why we need to send a control packet to update the connection's virtual status. An implementation of this VBR virtual status prediction can be found in the C++ simulator available at <http://www.ee.ubc.ca/~jeanf/pack.html>.

When all the information is known in the base station, that is, if there is no queued packet with non zero cell rate change, the virtual status in the base station is the same as the real status in the mobile. Otherwise, there is a difference between the virtual and real status. In these cases, over-allocation can occur when the base station allocates too many slots for the number of queued packet in the mobile. The mobile can use these slots to send update information as the one contains in control packets.

At the end of each frame the base station updates the virtual status prediction of each VBR connection based on the information received during the frame either piggybacked to data packets, in request packets or in over-allocated packets. Furthermore, virtual

packets for which the MTD is exceeded are removed from the virtual queues. Similarly, real packets are removed from the mobile's queues when their MTD is exceeded.

When the connections' virtual status have been updated in the base station, we allocate slots to VBR connections according to the virtual buffer status and connections QoS. The slot allocation algorithm is divided into two parts: guaranteed allocation and best-effort allocation. For the guaranteed service, VBR connections with packets in the virtual guaranteed queue are sorted in a list in increasing order of TTE where $TTE = MTD - (\text{Time} - \text{Arrival time of the first packet in the guaranteed queue})$. Let assume that A slots are available for VBR transmissions. The scheduler then allocates slots to the connections that have waiting packets in order of increasing TTE. The allocation algorithm can be summarized as follows:

while ($A > 0$)

 Allocate a slot to the first connection in the list

 Remove the first packet from the virtual guaranteed queue of the connection

$A = A - 1$

 Remove the connection from the list

if (There is a packet in the virtual guaranteed queue of the removed connection)

 Enter this connection in the list with the new TTE

Then, if $A > 0$, we allocate slots to the VBR connections with best-effort packets in their virtual queue. Based on the current best-effort virtual buffer length of all active VBR connections (an active VBR connection for the best-effort allocation algorithm is defined as a VBR connection with best-effort packets in its virtual buffer), the base station scheduler allocates the A slots that are still available to the connections according to the fair bandwidth allocation algorithm presented in Section 4.3.2.

When the allocation for guaranteed and best-effort packet is over, the base station announces the slot allocation to mobiles. However, the base station only indicates the VBR connections for which the allocation is made and not if the allocated slots are for guaranteed or best-effort packets. Each VBR connection uses its allocated slots to serve first all its queued guaranteed packets and then the best-effort packets. Since there might be a difference between the virtual and real status, the transmitted packets are not necessarily the one that the base station was expecting. This is why we need a field indicating the cell type and the sequence number in order to correctly update and predict the virtual status.

4.3.4 Algorithm for Contention Period

The problem for the uplink contention period can be divided into two parts: the protocol used to access the control slots and the number of slots to allocate to uplink control purpose in a frame. The contention access is controlled by the FPBP protocol presented in Section 3.3.2 that implements a random access protocol with mixed priorities. We will now describe the FPBP algorithm in the context of the DR-TDMA WATM MAC protocol.

Suppose that K^t control mini-slots are available in frame t and there are p different priority classes with arrival processes of intensities $\lambda_1, \dots, \lambda_p$. λ_i is the average number of control packet arrivals of class i per frame in all mobiles (it can be computed using a moving time-average of successful number of uplink control packet transmissions from class i per frame). A lower index corresponds to a higher priority control packet class. Let γ_i be the priority parameter of each control traffic class i . In order to maintain the priority order we must have $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_{p-1} \geq \gamma_p$ and the parameters satisfy the relation $\sum_{i=1}^p \gamma_i = 1$.

The algorithm operates by maintaining for each priority class i an estimate \hat{n}_i^t of the total number of backlogged control packets n_i^t at the beginning of each frame t . For each priority class i , an effective priority parameter $\hat{\gamma}_i^t$ is also computed. A new control packet arrived during frame t is immediately regarded as backlogged and it will attempt transmission in each subsequent frame after its arrival until success.

At the beginning of each frame t , for each priority class i , \hat{n}_i^t is updated from \hat{n}_i^{t-1} , $\hat{\gamma}_i^{t-1}$, K^{t-1} and the feedback for frame $t-1$ (let k_{nc} be the number of idle or success slots and k_c the number of collision slots in frame $t-1$) according to equation (3.44) and repeated here:

$$\hat{n}_i^t = \lambda_i + k_{nc} \max\left(0, \frac{\hat{n}_i^{t-1}}{K^{t-1}} - \hat{\gamma}_i^{t-1}\right) + k_c \left(\frac{\hat{n}_i^{t-1}}{K^{t-1}} + \frac{\hat{\gamma}_i^{t-1}}{e-2}\right) \quad (4.8)$$

After having computed \hat{n}_i^t , the estimated total number of backlogged uplink control packets at the beginning of frame t , for each traffic class i , a certain number of uplink slots will be requested for partitioning into uplink control mini-slots during frame t . Let CR be the number of control mini-slots per slot and CSmax the maximum number of slots that can be requested for control purpose. The later parameter is used to avoid the condition that too many slots are requested for contention and not enough slots are available for data traffic. Then, the number of uplink slots requested to be assigned to control purpose is determined as follows:

$$\text{Total} = \sum_{i=1}^p \hat{n}_i^t$$

if (Total ≥ 1)

$$\text{Request} = \min(\lceil \text{Total}/\text{CR} \rceil, \text{CSmax})$$

else

$$\text{Request} = 0$$

This ensures that when the algorithm estimates that there is at least one control packet waiting for transmission (i.e. backlogged), uplink control slots will be assigned in the

following frame. Another process is also used to allocate control slots when the VBR best-effort and ABR traffic are too heavy. Before allocating slots for best-effort traffic, if the time since the last allocation for control slot exceeds a parameter T_{max} , then a slot is assigned for control purpose in the next frame. This process assures that when there is no CBR, VBR guaranteed or voice packets to transmit, a contention period will be regularly available to allow potential time sensitive control packets to be transmitted. After slot allocation for CBR, voice, data and VBR guaranteed and best-effort traffic has been done, all unused slots are converted to control slots.

Once the estimated number of backlogged packets for each traffic class has been computed and K^t the number of control mini-slots in frame t is known, we therefore find the effective priority parameter of each class using the following prorating algorithm (see Section 3.2.2 for an explanation of the prorating algorithm):

for (each priority class i in order of increasing priority)

$$\hat{\gamma}_i^t = \min\left(\hat{n}_i^t/K^t, 1 - \sum_{j=1}^{i-1} \hat{\gamma}_j^t - \sum_{j=i+1}^p \min(\hat{n}_j^t/K^t, \gamma_j)\right)$$

$$L = 1 - \sum_{i=1}^p \hat{\gamma}_i^t$$

for (each priority class i)

$$\hat{\gamma}_i^t = \hat{\gamma}_i^t + \frac{\lambda_i}{\sum_{j=1}^p \lambda_j} L$$

Each backlogged packet is independently transmitted in the uplink control period of frame t according to the transmission probability q_i^t of the priority class i it belongs to. Transmission probabilities are calculated in the base station as follows:

$$q_i^t = \min\left(1, \frac{\hat{\gamma}_i^t}{\hat{n}_i^t} K^t\right) \quad (4.9)$$

If a packet is transmitted in a given frame, it will independently choose a given uplink control mini-slot for transmission with an uniform probability (each mini-slot has a probability $1/K^t$ of being chosen).

4.3.5 Traffic Integration

Until now, we have described how the allocation is independently done for each type of traffic. However, the different WATM services share the same resources and there must be an interaction between the allocation algorithms. Figure 4.3 illustrates the algorithm used to integrate the different types of traffic for uplink transmission in the DR-TDMA protocol. As we can see, the available slots are distributed first to CBR traffic, second to requested contention slots, then to voice and VBR guaranteed traffic and finally to data and VBR best-effort traffic. If there is any leftover, it will be assigned to control slots. This order of allocation respects the different priorities and QoS of each ATM service.

Although we have not specified an allocation algorithm for CBR traffic, in each frame slots are granted to CBR connections according to the bit rate specified at the CBR connection admission. A scheduling algorithm similar to the one proposed for voice traffic in Section 4.3.1 can be used. CBR traffic, like voice traffic during a talkspurt, has a constant bit rate from which the next packet arrival time can be predicted. CBR connections can thus be served using a Time-To-Expiry scheduling algorithm. When a new CBR connection is established, the mobile sends a high priority request packet to the base station using the FPBP protocol. The request packet contains the requested connection bit rate and, if the connection is admitted, the base station add the connection to the list of CBR connections. CBR connections in the list are served according to the TTE scheduling algorithm. A CBR connection is only removed from the list when it is tear down.

The allocation for voice and VBR guaranteed traffic is made together since their allocation algorithms are very similar. Actually, we maintain a single list of the active voice connections and VBR connections with queued virtual guaranteed packets in increasing order of TTE. Connections are then allocated slots in order of increasing TTE. This

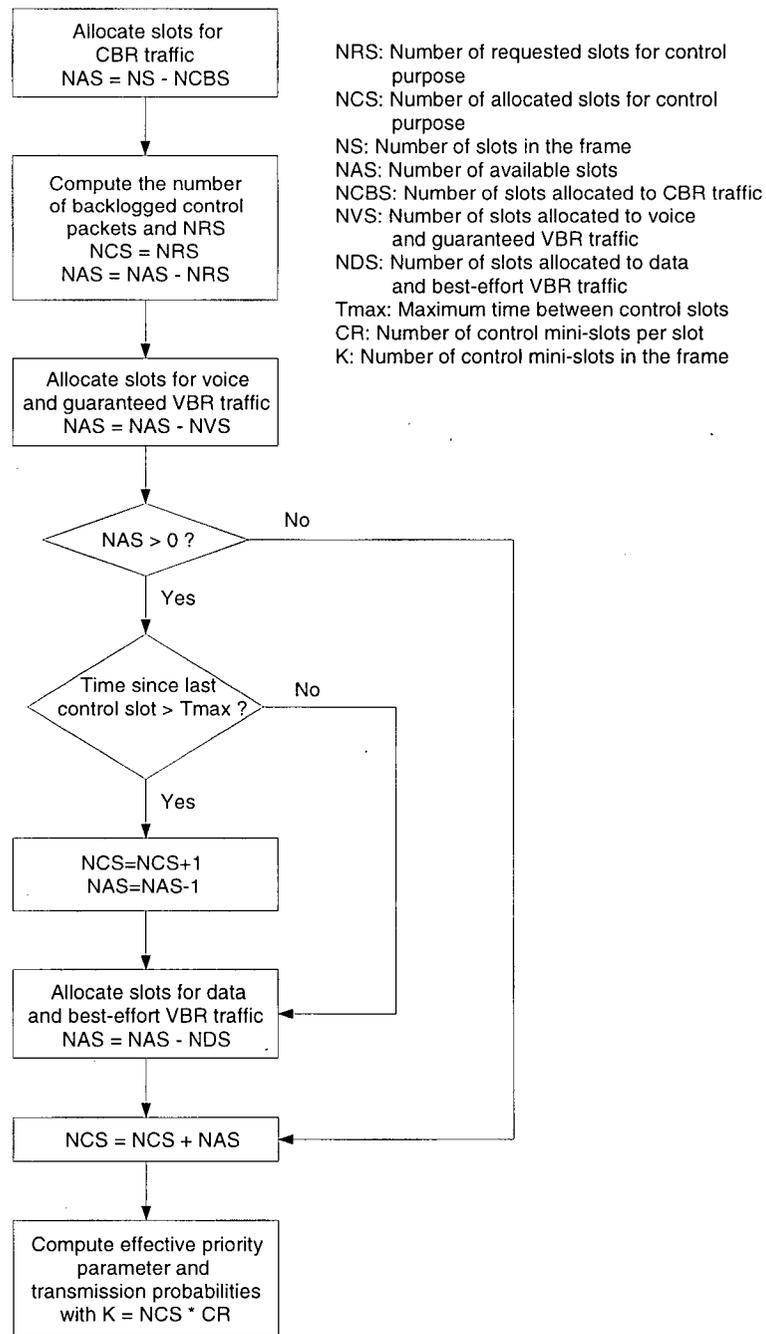


Figure 4.3: Traffic Integration Flow Chart

allows guaranteed VBR packets and voice packets to receive the same quality of service according to their traffic parameters.

Similarly, data and best-effort VBR allocation algorithms both use the fair bandwidth allocation algorithm. It is thus easy to integrate these allocation algorithms together. We thus maintain a single list containing the active data connections and the VBR connections with queued virtual guaranteed packets. The fair bandwidth allocation algorithm is then executed with this list.

The current NAS is the number of available slots A used at the beginning of the Time-to-Expiry and fair bandwidth allocation algorithms. For control slots, the number of backlogged control packets at the beginning of the frame, and accordingly the number of requested uplink control slots, is computed using the feedback from the previous frame control slots. After the allocation for CBR, voice, VBR and data traffic is done, the number of control slots available in the current frame is known. Then, the effective priority parameter and the transmission probabilities are computed. When the integration algorithm is finished, the slot allocation is announced in downlink control slots. Contention parameters (number of control slots and transmission probabilities) are also transmitted.

4.3.6 Discussion on the DR-TDMA Allocation Algorithms

In this section, we discuss the general properties of the proposed DR-TDMA scheduling algorithms. Performance results of the protocol are presented in Section 4.4. An important aspect of the allocation algorithms is the computation power that they require. The algorithms are inherently simple, however they suffer from a problem of scalability. That is, the computation power that they require increases when the number of connections increases. This is due to the fact that for each connection, we have to independently maintain status information about the connection state. Furthermore, for

the TTE algorithm, when the number of admitted connections increases, the length of the list containing the active connections increases. Therefore, the management of the list order requires more computation. Similarly, when the number of data and VBR connections increases, the fair bandwidth allocation algorithm requires more computational effort. On the other hand, the computation of control packets transmission probabilities and of the number of requested control slot is simple and is independent of the number of connections.

While the slot allocation for CBR, VBR and data connections must be computed at the end of the frame, some operations can be done at other moments. For example, status prediction for voice and VBR connections can be computed at any time. Furthermore, since we have predict the connections status, the TTE list ordering can also be done before the end of the frame. Only for connections for which new state information has been received in the current frame will there be a need to recompute the status prediction at the end of the frame and reorder them in the TTE list if necessary. Similarly, the estimates of the numbers of backlogged control packets and the number of requested control slots can be computed immediately after the contention period. By performing parts of the allocation algorithms during the frame, we can reduce the burden at the end of the frame and therefore decrease the required computing power.

In order to give an idea of the processing power required for the allocation algorithms, we can examine the CPU time required to process a simulation run. For example, simulation run of the integrated allocation algorithm in Section 4.4.4 with 100 voice connections, 100 data connections and 15 VBR connections during a period of 5000 seconds, requires around 4000 seconds of CPU time. The simulator has been programmed in C++ and was run on an UltraSparc II-200MHz computer with a Solaris 2.5.1 system, which is about 12 time faster than a Sparc 10 computer. However, we should note that the required CPU also includes computational efforts to generate the traffic, to perform management

functions in the mobiles and to collect statistics. Therefore, we can conclude that the allocation algorithms can be practically implemented in a base station for a real time wireless ATM network.

The DR-TDMA allocation algorithms performance is sensitive to the number of control mini-slots per slot. That is, when the number of control mini-slots per slot increases, less slots must be assigned to control purpose to obtain the same performance for contention traffic. There is therefore more slot available to carry data and thus increasing the overall performance of the DR-TDMA protocol (cell loss rate, throughput and delay performance). However, the ratio of 3 control mini-slots per slot that we chose for our protocol is conservative compare to previous studies: for example, in [4] they used 5 control mini-slots per slot while in [28] and [29] a ratio of 7 control mini-slots per slot is selected.

For voice, data and VBR allocation algorithms we use piggybacked bits in data packets to carry information about the connection state. This makes the base station prediction algorithms sensitive to packets losses. However, this would not affect the operation of the algorithms, but just the short term accuracy of the status prediction. Packets corrupted in the radio channel are eventually retransmitted (if their MTD is exceeded the mobile will send a control packet) and the prediction can be updated.

Piggybacked bits reduce the net utilization of data slots without this being reflected in the results. We can however estimates the number of overhead bits per data packets that will be required for piggybacked bits. For voice packets, only one bit is needed to indicate if the packet is the last of a talkspurt. In VBR packets, piggybacked bits are used to indicate the packet type (best-effort or guaranteed packet) and the rate-level change (coding the rate change instead of the actual rate level reduces the number of required bits). One bit is thus needed to indicate the packet type while the number of bits required to code the rate change depends on the allowed variations of the source rate. If

we limit the variations of the source rate level (i.e. $l^{c+1} - l^c$) to an absolute value smaller than 16 for each transmitted packet, then 5 bits are enough to code the rate change. Furthermore, the limit on the allowed rate change still permits a relatively bursty VBR traffic flow. Therefore, we can estimate that 6 bits are required to piggyback information for the VBR allocation algorithm. In ABR packets, piggybacked information is used to indicate the number of packets in new batch arrivals. With 6 bits we are limited to a maximum batch size of 63 packets. However, if the number of packets in a batch arrival is larger than 63 packets, for the allocation algorithm we can split the new batch arrival into several batch arrivals of size smaller than 63 bits. We can therefore piggyback the information on multiple data packets.

Thus, we can estimate that 6 bits of overhead per WATM cells are required for piggybacked bits. This represents a 1.25% ($6/(60*8)$) reduction of the net data slot utilization. This is quite small compared to the estimated 20% overhead for wireless ATM packets (see Section 2.5.1). For control packets, the required payload for allocation algorithms is much more difficult to estimate (for example it depends on the time granularity, sequence numbering, ...). However, if the 64 bits per control packets are not enough to transmit the control information, several control packets can be used.

4.4 Simulation Results

A C++ simulator has been written to evaluate the performance of the proposed wireless ATM MAC protocol. Because we strictly focus on the MAC aspect, we have assumed a perfect radio channel without errors and fading. The simulations were run for a minimum simulation time of 5000 seconds in order to assure accurate results. We have simulated the protocol with the system parameters given in Table 2.1. In order to see

Table 4.2: Frame parameters for simulations

Slots per frame	5
Frame duration	4 ms
Channel bit rate	644 Kbps

the influence of the channel data rate on the protocol, we also simulated it with 5 slots per frame. The parameters for this channel are given in Table 4.2. For the FPBP access protocol, the number of priority levels p is equal to 2. Voice and VBR control packets are assigned priority 1 and data control packets priority 2. We chose the optimal values of $\gamma_1 = 1$ and $\gamma_2 = 0$ for the priority parameters. The length of the window used for the moving time-average of the number of successful control packet transmissions for each traffic class is set to 100 frames. For most of the simulations CSmax has been set to 2 and Tmax to 12 msec. We have assumed an infinite buffer model where cell losses are only due to exceeding the MTD.

We only intend to evaluate the efficiency of the protocol for uplink transmissions, therefore we have assumed that downlink control slots take no transmission bandwidth. The throughput is defined as the ratio of the average number of slots used for packet transmissions per frame (excluding control packets) to the total number of slots available per frame. The offered load thus includes cells headers but does not include control packets. Therefore, control traffic is considered as a bandwidth loss in the throughput calculation. The throughput is thus given by:

$$\text{Throughput} = \frac{\text{Average number of slots allocated to user traffic per frame}}{\text{Number of slots per frame}} \quad (4.10)$$

where the number of slots per frame is the parameter given in Tables 2.1 and 4.2.

Furthermore, in the simulator we have not considered the exact position of allocated slots inside a given frame. This just marginally affects the delay experienced by a packet.

In a real implementation, slot positions will be chosen such that packets that will first exceed their MTD will be allocated the first slots in the frame. In the simulation model implementation we have considered that packet transmissions all take place at the end of the frame. This decreases the simulation time and barely affects the results. Moreover, this approximation is reasonable since we do not know exactly the processing time and delay tolerance values. For example, suppose we consider a voice packet that has a MTD of 16 ms. We might find that the packet has experienced exactly a 16 ms delay. However, we must add the processing delay (which we do not know). Furthermore, the packet must go through the wired network and perhaps through another wireless network before being delivered to the receiver. What is critical for the voice packet is the end-to-end delay. Therefore, a packet having experienced a delay of 16.2 ms in the wireless ATM network might also be good. This is why we believe that using the exact departure time does not give a better insight on the protocol performance.

We first evaluate the protocol with voice connections only and with VBR connections only. We then integrate the voice and data traffic to specifically evaluate the FPBP protocol. Finally, the results of a complete integration of voice, data and VBR traffic are presented.

4.4.1 Results for Voice Traffic Only

First, we simulate the MAC protocol with the channel speed of 8.528 Mbps. Then to see how the protocol works under low channel speed condition, we also evaluate the performance with a channel speed of 664 Kbps. Furthermore, for comparison purposes, we also evaluate the ideal reservation protocol where the control packet at the beginning of a talkspurt immediately reaches the scheduler without using any bandwidth. This ideal system behaves like a perfect multiplexer and thus gives the performance upper

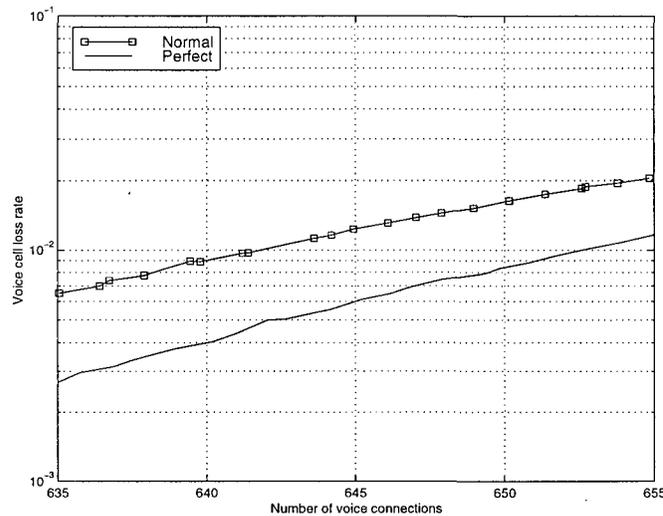


Figure 4.4: Voice loss rate as a function of the number of voice connections ($R = 8.528$ Mbps).

bound. Figures 4.4 and 4.5 shows the voice cell loss rate and the throughput obtained with a channel speed of 8.528 Mbps.

Some studies suggest that the voice quality degradation due to packet losses is only perceptible when the cell loss rate exceeds 1% [21]. Using this criterion, for a channel speed of 8.528 Mbps the perfect system can sustain 653 simultaneous voice connections while the proposed system supports 643 sources (Figure 4.4). Recall that the voice source activity factor is 0.4255 (i.e. $t_b^{vo}/(t_b^{vo} + t_i^{vo}) = 1/2.35$) and one voice packet is generated each 16 ms. Thus, if the statistical multiplexing was perfect, a voice source would request on average 0.4255 slot per 16 ms, such that the channel could support a maximum number of 658 simultaneous sources ($35 \cdot 8 / 0.4255$). However, due to statistical variations, the multiplexing can not be perfect and, for this channel speed, we have a statistical multiplexing loss of 5 sources as indicated by the performance of the ideal system (653 sources). Instead of using 0.4255 slot per 16 ms, each source needs 0.4288

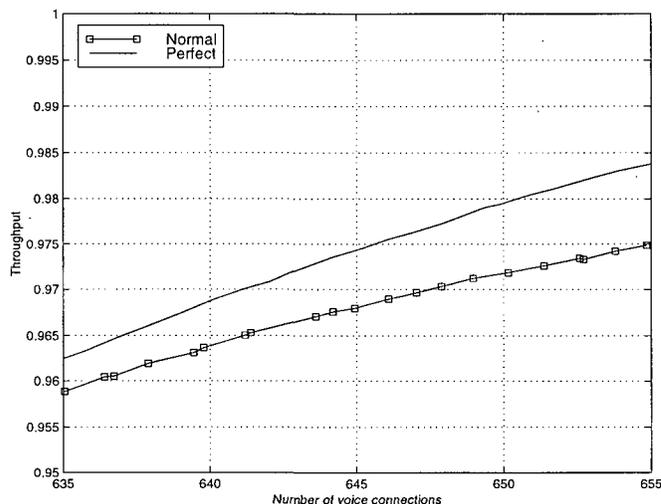


Figure 4.5: Throughput as a function of the number of voice connections ($R = 8.528$ Mbps).

slot. For the “real” system, there is an additional loss due to the transmission of requests at the beginning of talkspurts. This traffic causes a loss of 10 sources (i.e. a 1.5% degradation), that means that 4.29 slots ($10 \cdot 0.4288$) per 16 ms are lost because of the request traffic. We can estimate that on average there will be 0.547 request per frame (i.e. $1/(t_b^{vo} + t_i^{vo}) \cdot \text{Frame length} \cdot \text{Number of connections} = 1/2350 \cdot 2 \cdot 643$). The contention system requests a control slot each time it estimates that there is a waiting control packet. Thus there will be a request for a control slots after two frames (i.e. 4 ms) or 4 requested control slots per 16 ms which approximately corresponds to the loss of 10 voice sources.

The previous estimation is based on the assumption that there is no collision. In fact the contention throughput is $0.368 (1/e)$ and there are 3 control mini-slots per slot. During a period of 16 ms, exactly 6.38 slots on average are not used by voice traffic ($35 \cdot 8 - 643 \cdot 0.4255$). If the control mini-slots utilization was perfect, 3.97 slots per 16 ms on average would be needed ($0.547 \cdot 8 \cdot e/3$). Therefore, if everything in the real

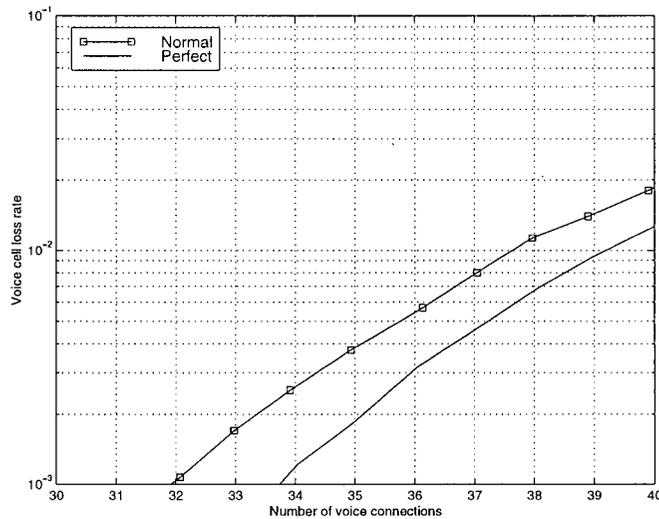


Figure 4.6: Voice loss rate as a function of the number of voice connections ($R = 664$ Kbps).

system was perfect, only 5 more sources could be supported $((6.38-3.97)/0.4255)$ which represent less than 1% of the number of sources that can be admitted. This points out the efficiency of the contention protocol.

We can further observe that at a 1% loss, the throughput is equal to 96.75%, which means that 96.75% of the slots are used to transmit voice packets, while the others are used for control packet transmissions or are unused. Previous studies have shown that traffic congestion in the control channel is a throughput limiting factor for packet reservations over a high speed channel [24]. However, our results prove that our protocol can achieve near perfect channel utilization in a high speed environment.

For the slower channel, Figure 4.6 shows that the statistical multiplexing gain decreases with the channel speed: a voice source needs 0.526 slot (20/38) per 16 ms for a channel speed of 664 Kbps. However, the performance of the proposed protocol is still close to the perfect system performance. For the 664 Kbps bit rate channels, we can

Table 4.3: VBR connections parameters

Number	Connection Bit Rate	Basic source cell interarrival time ($1/R_b$)	Number of ON/OFF sources (S)	Average ON state duration (t_b^{vbr})	Average OFF state duration (t_i^{vbr})
#1	250 Kbps	7.68 ms	15	100 ms	200 ms
#2	250 Kbps	7.68 ms	15	1000 ms	2000 ms
#3	667 Kbps	3.6 ms	25	100 ms	300 ms

compare our system with the PRMA protocol presented in [65] where similar parameters have been used. In the PRMA system, 34 simultaneous voice sources can be supported while our system can accept 38 sources which represents an improvement of 12%. In [24], the voice model is different and they used a polling system. However, with 20 slots per 16 ms period (which corresponds to our speed of 664 Kbps) they observed a loss of 2 voice sources with respect to the perfect system. The performance of the DR-TDMA protocol is slightly better. However, for high channel speeds, they observed a performance degradations while our system remains stable and efficient.

4.4.2 Results for VBR Traffic Only

Table 4.3 presents the parameters of the different VBR connections used in our simulations. For all the VBR connections we have set the MTD to 50 ms. The parameters of the 667 Kbps connection are similar to the traffic characteristics in [29]. Although they used a compressed digital video sequence, we have set our model such that the traffic parameters (average bit rate and peak-to-average ratio) are similar. For the cell loss rate and delay performance measurements, we present the results with the statistics computed either only for guaranteed packets or for the total traffic. In the cases where g is equal to the number of ON-OFF sources S , the two statistics are the same since all

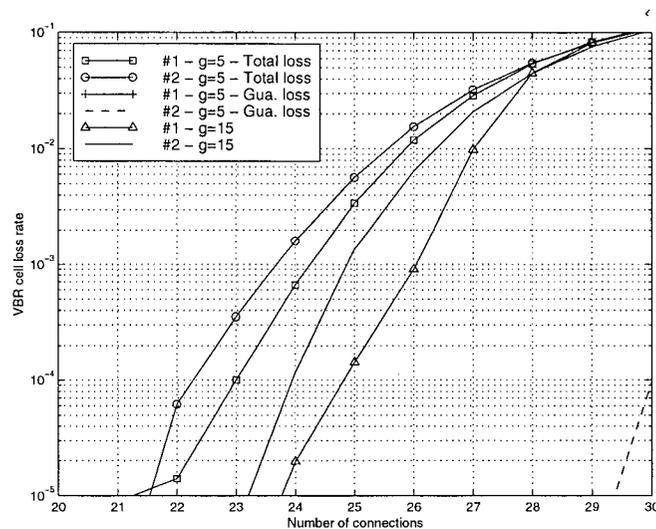


Figure 4.7: VBR cell loss rate as a function of the number of VBR connections (connections #1 and #2).

the packets are of the guaranteed type.

Figures 4.7 to 4.9 present the simulation results for the VBR cell loss rate, the VBR cell delay and the throughput obtained for the 250 Kbps connections (connections #1 and #2). Figures 4.10 to 4.12 present the same results for the 667 Kbps connection (connection #3). In these simulations, for each type of VBR connections, we have simulated the MAC protocol with different values of the guaranteed parameter g in order to observe the impact of the SCR on the performance. We have set the token pool depth W to one token (the effect of the token pool depth on the performance will be analyzed later).

From the cell loss rate results (Figures 4.7 and 4.10), we see that for all simulated conditions the total loss rate remains low (below 1%) for an offered load below 93%. What is interesting to note is that the connection speed only slightly affect the cell loss performance (for example when all the VBR traffic receive a guaranteed service, with either 250 or 667 Kbps VBR connections we have a 1% loss at about a 97% throughput).

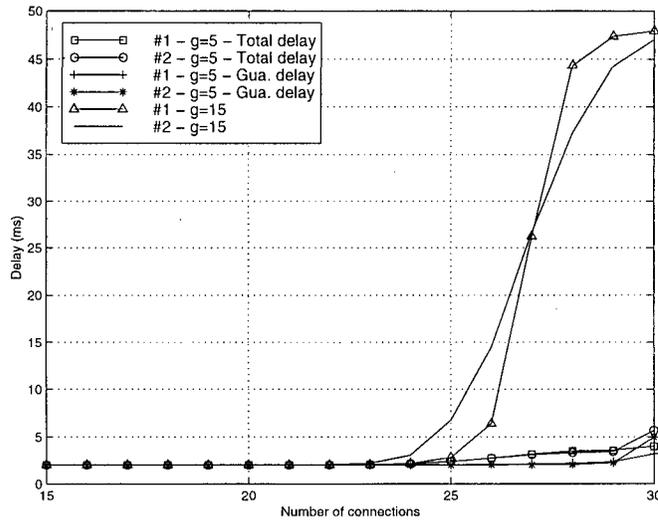


Figure 4.8: VBR cell delay as a function of the number of VBR connections (connections #1 and #2).

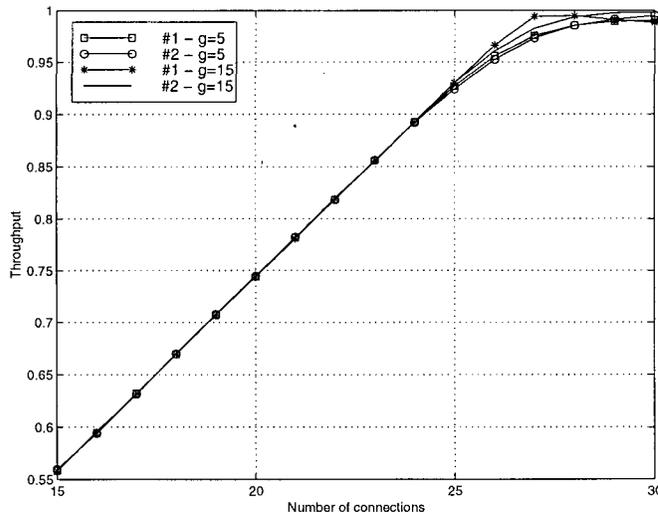


Figure 4.9: Throughput as a function of the number of VBR connections (connections #1 and #2).

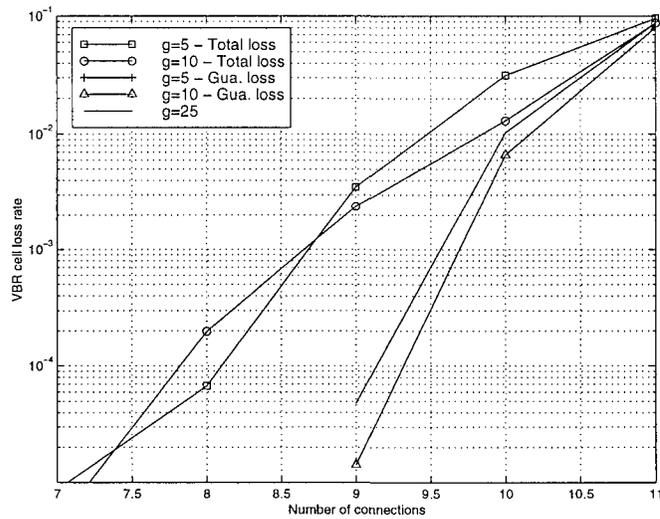


Figure 4.10: VBR cell loss rate as a function of the number of VBR connections (connection #3).

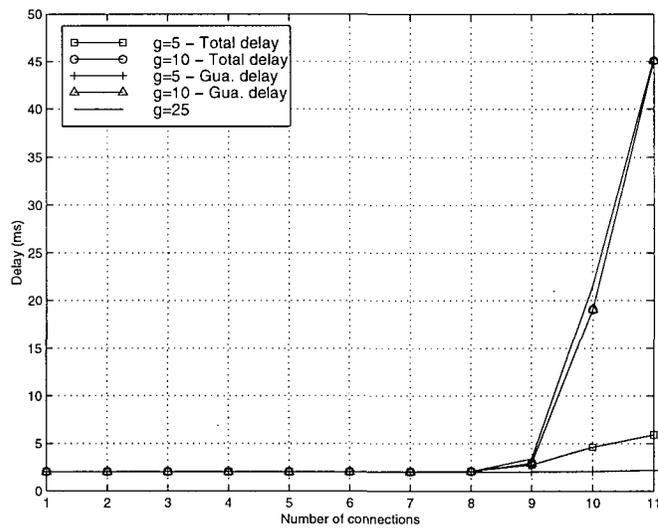


Figure 4.11: VBR cell delay as a function of the number of VBR connections (connection #3).

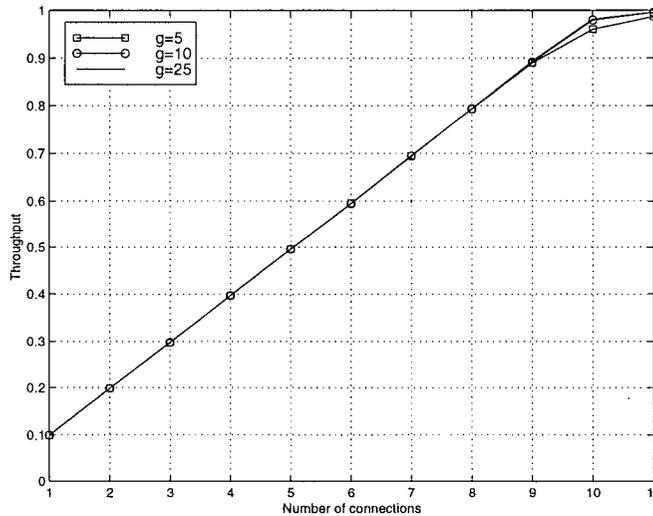


Figure 4.12: Throughput as a function of the number of VBR connections (connection #3).

Furthermore, we can observe that when we decrease the parameter g , the total loss rate slightly increases but, as we expected, there is a significant improvement of the guaranteed cell loss rate. For small values of g we often get a zero guaranteed cell loss rate. This confirms that our protocol is able to deliver a better quality of service to guaranteed traffic.

Similar conclusions can be drawn from the delay performance of the protocol (Figures 4.8 and 4.11). What is interesting to note is that when we decrease the parameter g , the delay performance improves for the total and guaranteed traffic. This can be explained by the fact that the best effort loss rate increases when g decreases, therefore the packets that experienced a longer delay are more often discarded and are not included in the waiting time statistic.

The concept of guaranteed cell rate to allow a fraction of the total traffic to receive a better quality-of-service clearly affects the performance of the high priority traffic.

When the guaranteed cell rate (i.e. parameter g) is increased, the total multiplexing performance improves as shown by the amelioration of the total loss rate. However, the quality-of-service offered to guaranteed traffic decreases as g increases. On the other hand, when the guaranteed cell rate is decreased, the guaranteed traffic QoS improves but it causes a performance decrease of the total traffic since a higher portion of the traffic receives a best effort service. We can improve the performance of the best effort service by including in the fair bandwidth algorithm a mechanism to take into account the TTE of the best-effort packets. Furthermore, when the guaranteed cell rate is decreased, it allows a better access for data connections since best effort and data packets receive the same quality-of-service. Finally, it should be noted that the implementation of the guaranteed traffic mechanism implies a complexity increase in the mobile to manage both queues and in the base station for the virtual status maintenance algorithm.

When we increase the ON and OFF state durations for the 250 Kbps connection (connection #2), we observe in Figures 4.7 and 4.8 a slight decrease in the performance: the delay is longer and we have a higher loss rate. This phenomenon should be expected since when we increase the state length, the statistical multiplexing is less effective. However, what is interesting to note is that the performance deterioration is small even for such a great change in the average state length. This shows that our protocol can perform well for a wide variety of connections.

Another useful performance measure introduced in [29] is the *Allocation Efficiency* (AE) defined as the fraction of slots allocated to VBR traffic that are actually used for transmission. The AE index indicates the match between the virtual status and the real status. Furthermore, a higher AE will allow the system to support more data traffic. For all the simulated conditions we have obtained AE values superior to 99% and most of the time over 99.8%. This indicates the efficiency of our prediction algorithm.

For purpose of performance comparison, the connection and channel parameters used

for the simulations with the 667 Kbps VBR connection at a channel speed of 8.528 Mbps have been set similar to the one used for the NEC's protocol in [29]. Figures 4.11 and 4.12 can thus be compared with the results presented in [29]. At low throughput, the DR-TDMA and NEC protocols both show a stable delay around 2 ms. However, if we take a target delay of 5 ms, the maximum throughput of the NEC protocol is around 64% while we can attain a throughput of 90% with the DR-TDMA protocol. Similarly, if we target a 10 ms delay, the maximum throughput that the NEC protocol can achieve is 72% while our DR-TDMA protocol can reach between 95% and 99% throughput depending on the value of g . For the AE performance, our DR-TDMA protocol gives a constant value of 99.1%, while the NEC protocol has an AE varying between 75% and 80%. From these results, we can see that our DR-TDMA MAC protocol clearly outperform the NEC's bandwidth allocation algorithm proposed in [29]. This can be explained by the efficiency of our prediction algorithm (as proved by the AE results) and the use of the dynamic FPBP protocol to control the access to the contention slots and to select the number of uplink control slots.

We have also simulated the protocol with a lower channel speed of 667 Kbps with connections bit rates of 26.6 and 50 Kbps. As expected the statistical multiplexing gain is lower but we still have a good performance. For throughput in the range of 85 to 90%, the cell loss rates still remain under the 1% criterion. Similar results as the ones obtained for the high speed channel are observed for the delay and AE performance. The impact of the guaranteed cell rate on the protocol performance is similar.

In the previous simulations, we have used a guaranteed token pool depth W of one token. The parameter W allows a better adjustment of the guaranteed traffic characteristics. When W is set to one, no bursts of guaranteed traffic is allowed, while when W increases traffic bursts are tolerated in the guaranteed traffic. We have therefore compared the performance of the system with different value of W . We have simulated

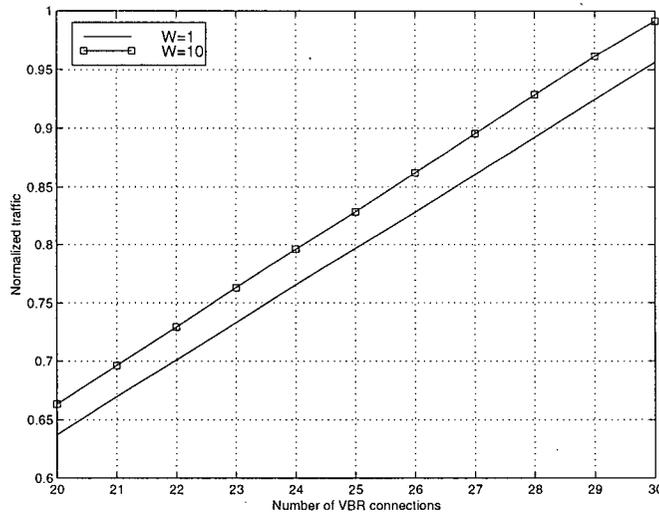


Figure 4.13: Comparison of the normalized guaranteed traffic load as a function of the number of VBR connections for different values of W .

the DR-TDMA protocol with the parameter of the VBR connection #1 presented in Table 4.3. We have used a sustainable guaranteed cell rate parameter g of 5 and we have compared the cell control algorithm with token pool depth W of 1 and 10 tokens. Figure 4.13 shows the normalized VBR guaranteed traffic load for these two values of token pool depth. We can observe that since, with a greater value of W , bursts are allowed in the guaranteed traffic, the guaranteed traffic load slightly increases by about 3% when $W = 10$. This has a small impact on the experienced total cell loss rate as illustrated in Figure 4.14. With $W = 10$ the guaranteed traffic is higher, therefore the total cell loss rate slightly improves while the guaranteed cell loss rate increases similarly to what was observed previously when g was increased. While the impact of the guaranteed token pool depth value on the cell loss rate is small for these traffic conditions, we should recall that the goal of the parameter W is to offer more flexibility on the definition of the guaranteed traffic characteristics as illustrated by Figure 4.13. The ON-OFF model used

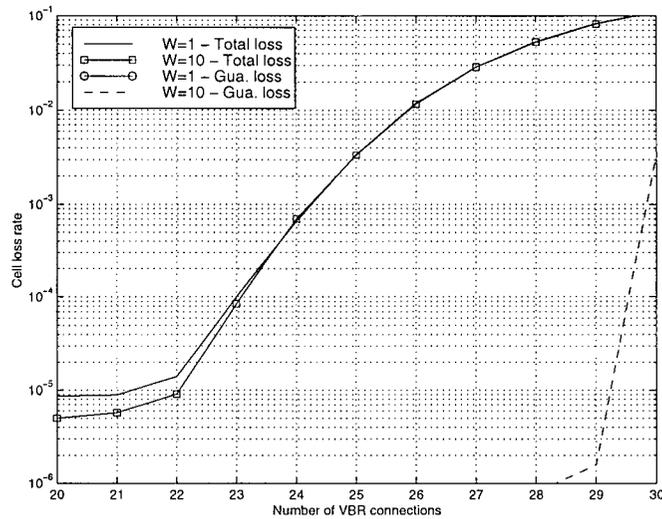


Figure 4.14: Comparison of the cell loss rates as a function of the number of VBR connections for different values of W .

to characterized the VBR sources is known to be slowly varying which can explain the small impact of W . For burstier traffic, the flexibility that W offers to characterize the maximum burst length can help the user to improve its QoS by better defining its VBR traffic.

4.4.3 Results for Data and Voice Integrated System

In this section, we are concerned with the system performance when voice and data traffic are integrated together on the wireless ATM channel. The three following measures will help to evaluate the system performance and efficiency:

- Voice cell loss rate;
- Data waiting time;
- Channel throughput.

Table 4.4: Parameters for data and voice integrated system simulations

Channel Speed	Number of voice connections	t_d	β	μ_b	Data source bit rate	CR	CSmax
8.528 Mbps	100	0.1 s	3	5	19.2 Kbps	3	2
8.528 Mbps	50	0.1 s	3	10	38.4 Kbps	3	2
644 Kbps	25	0.2 s	3	3	5.76 Kbps	1	1

We have first evaluated the proposed system with the 8.528 Mbps channel parameters presented in Table 2.1. We have also performed other simulations to evaluate the protocol with the 644 Kbps channel presented in Table 4.2. Table 4.4 presents the data source model and contention algorithm parameters used in the simulations. For a definition of the parameters see Sections 4.2.2 and 4.3.4. We have set the maximum transfer delay for data packets to 60 seconds. For each set of parameters, we have simulated the “Non-Priority” system where voice and data control packets are assigned the same contention priority and the “Priority” system where voice and data control packets respectively have a high and low contention priority. Therefore, we can evaluate the impact of the proposed FPBP protocol on the DR-TDMA WATM MAC protocol.

Figures 4.15 to 4.23 present the simulation results. From the cell loss results (Figure 4.15, 4.18 and 4.21), we can observe that the priority scheme significantly reduces the voice cell loss rate when it becomes too high without the priority protocol. It has to be noted that for certain set of parameters the voice loss rate can be low with the non-priority system and therefore, the impact of the FPBP protocol is less significant since there is nothing to improve. Still, the presented results show that if the cell loss rate would become high enough to degrade the voice connections performance, the FPBP protocol keeps it to an acceptable level where the cell loss rate is not a limiting factor of the

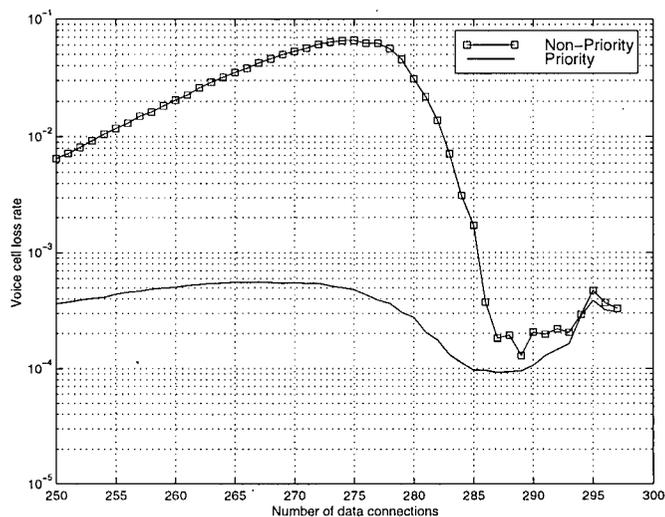


Figure 4.15: Voice loss rate as a function of the number of data connections with 100 voice connections ($R = 8.528$ Mbps).

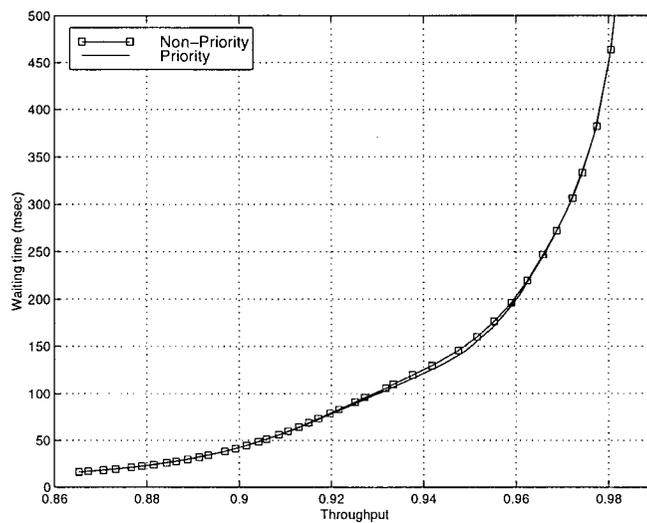


Figure 4.16: Data waiting time as a function of the throughput with 100 voice connections ($R = 8.528$ Mbps).

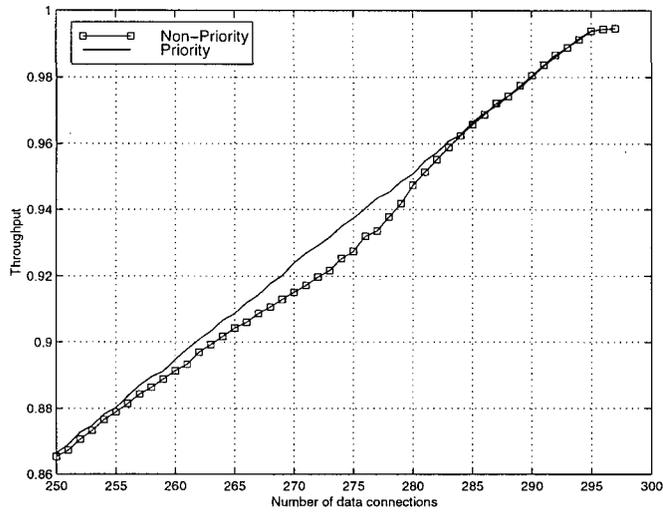


Figure 4.17: Throughput as a function of the number of data connections with 100 voice connections ($R = 8.528$ Mbps).

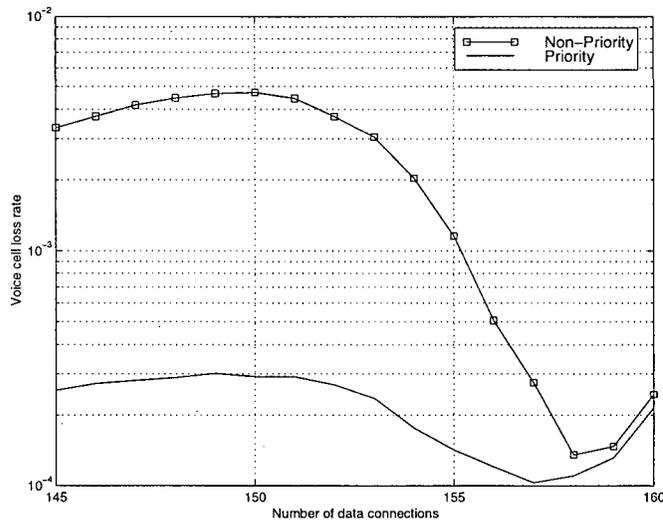


Figure 4.18: Voice loss rate as a function of the number of data connections with 50 voice connections ($R = 8.528$ Mbps, mean batch length: 10 packets).

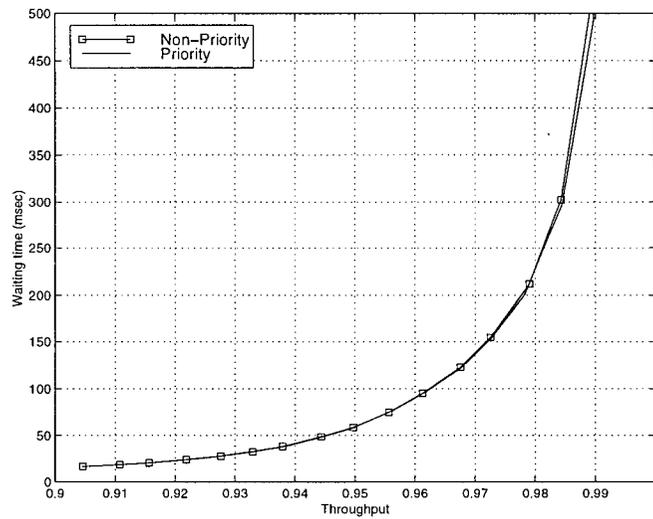


Figure 4.19: Data waiting time as a function of the throughput with 50 voice connections ($R = 8.528$ Mbps, mean batch length: 10 packets).

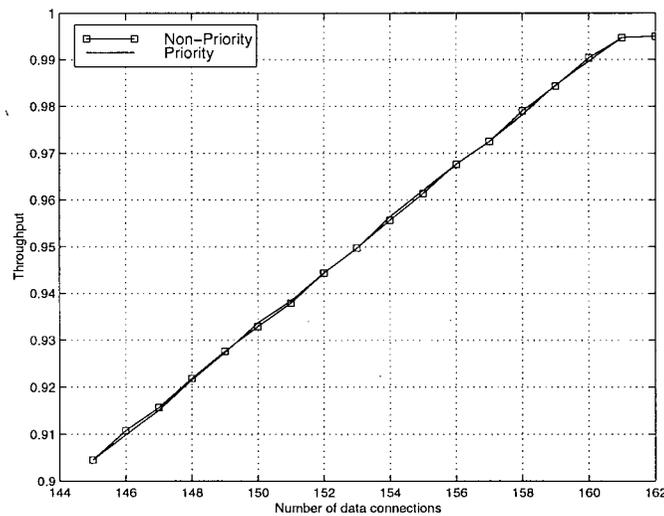


Figure 4.20: Throughput as a function of the number of data connections with 50 voice connections ($R = 8.528$ Mbps, mean batch length: 10 packets).

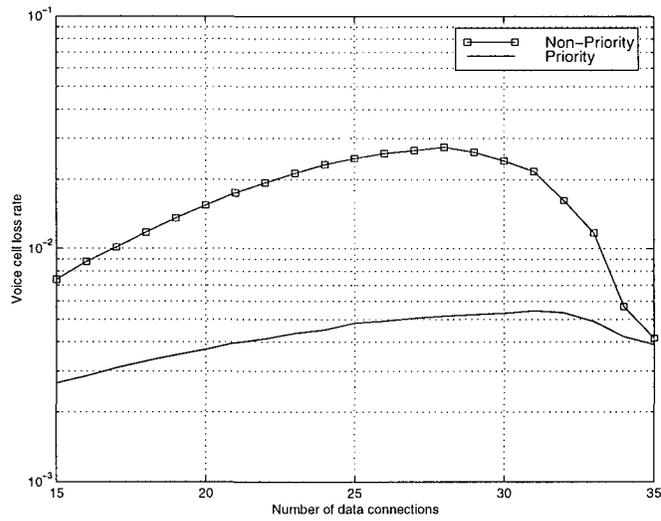


Figure 4.21: Voice loss rate as a function of the number of data connections with 25 voice connections ($R = 664$ Kbps, $CR = 1$, $CS_{max} = 1$).

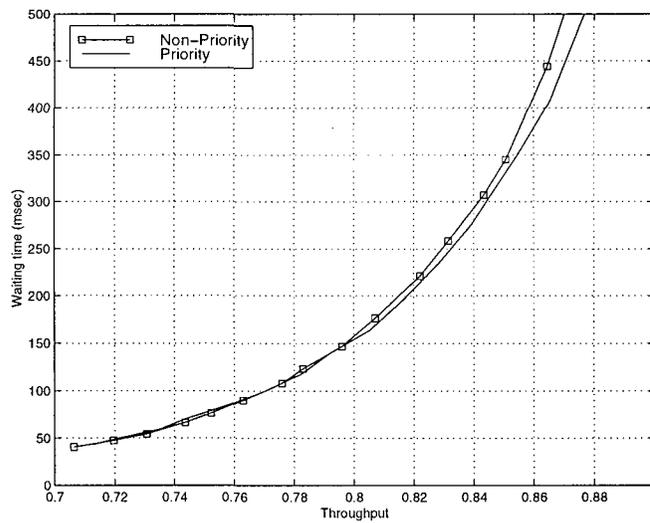


Figure 4.22: Data waiting time as a function of the throughput with 25 voice connections ($R = 664$ Kbps, $CR = 1$, $CS_{max} = 1$).

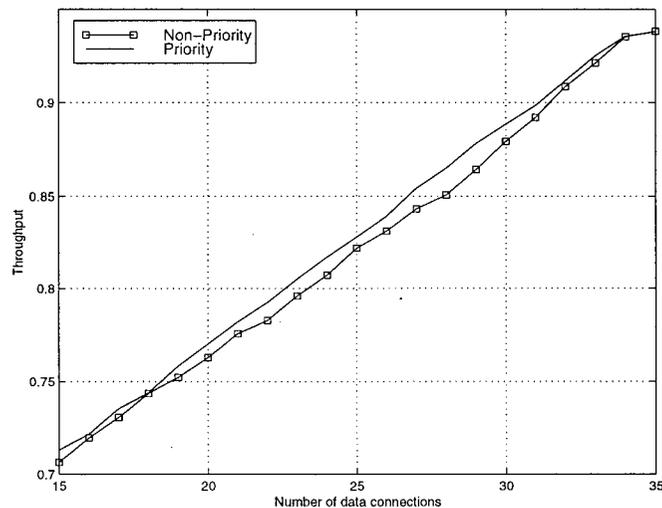


Figure 4.23: Throughput as a function of the number of data connections with 25 voice connections ($R = 664$ Kbps, $CR = 1$, $CS_{max} = 1$).

DR-TDMA MAC protocol. Furthermore, the voice packet loss rate performance improvement due to the FPBP protocol barely affects the data waiting time (Figure 4.16, 4.19 and 4.22) and throughput performance (Figure 4.17, 4.20 and 4.23).

The bell shape of the voice cell loss rate function can be explained by the effects of two different phenomena. When we increase the number of data connections, the traffic load and the number of control packets increase. There is thus less control slots available to carry a higher control traffic. The contention traffic thus increases causing a longer waiting time for control packets (for both data and voice connections) and therefore increasing the voice cell loss rate. However, while we increase the number of data connections, the data waiting time increases. Thus, a larger number of data request are piggybacked to data packets. The control traffic then declines which explains the decrease in the voice cell loss rate after a certain point.

From the throughput results (Figure 4.17, 4.20 and 4.23), we can observe that the maximum sustainable throughput is quite high for the 8.528 Mbps system. Depending on the parameters, it varies between 99% and 99.5%. That means that only 0.18 to 0.35 slots per frame are either used for control purpose or unutilized. The proposed DR-TDMA wireless ATM MAC protocol is thus quite efficient. However, under this load, the waiting time experienced by data packets is relatively high (we must underline that the voice quality always remains at an acceptable level with the priority system for any traffic load). There is no general agreement for the delay value over which the latency can cause some problems to upper layer like TCP or the delay that a user can tolerate. If we take the 250 ms delay introduced by a geo-stationary satellite hop as a reference, the sustainable throughput then varies between 96.5% and 98%. Thus our proposed protocol can support a high offered load (near the maximum that can be served by a perfect multiplexer) while maintaining the voice quality and the data waiting time at acceptable levels.

Even if the source models are not exactly the same, we can compare the DR-TDMA MAC protocol results with the performance of the NEC system presented in [28]. In the NEC MAC protocol, 12% of the slots are assigned for the uplink control period and 8% of the slots are assigned for the downlink control period (the maximum achievable throughput is thus limited to 80%). In our results, we have considered that downlink control slots take no transmission bandwidth, but, if we allocate 8% of the bandwidth to downlink control slots (the maximum achievable throughput is therefore limited to 92%), the DR-TDMA throughput is reduced from 96.5% to 88.8%. For the NEC protocol, the maximum achievable throughput is 75%. While the performance of the allocation algorithms for voice and data traffic are similar (i.e. achievable throughput versus maximum throughput), the dynamic nature of the DR-TDMA uplink control period as well as the FPBP algorithm allows the DR-TDMA protocol to require less uplink control slots and

achieve a much higher throughput than the NEC protocol.

When the channel speed decreases, as expected, the maximum sustainable throughput decreases. For the 644 Kbps channel, the maximum throughput is 93% while the throughput for a data delay of 250 ms is 83%. For this channel speed, the voice cell loss rate with the FPBP protocol always stays at levels where the voice quality is not affected, unlike the voice loss rate QoS when the FPBP protocol is not used.

Finally we can compare the DR-TDMA protocol with voice and data integration in the DRMA protocol presented in [4]. They use a voice model similar to the one that we selected in this thesis. However, in their data model each user generates independently a packet in each frame with a probability p_0 and is forbidden to generate a packet when one is backlogged in the mobile. We have implemented this model in our simulator and we have also used a modified version of this model where the mobile is not forbidden to generate a packet when one is backlogged (piggybacked request transmissions are used in this modification when there is already backlogged packets in the mobile). We have set the simulation parameters to values similar to the DRMA simulation conditions. Each simulation run is done with the same number of voice and data connections. Table 4.5 compares the simulation results obtained with the DR-TDMA and the DRMA protocol. Results are quite similar for the DR-TDMA and DRMA protocols. We can observe that the DR-TDMA protocol can provide a slightly better QoS for voice connections. However, it causes in a service deterioration for data traffic. But when we allow multiple backlogged packets in a mobile and the use of piggybacked request transmission, the DR-TDMA protocol offers similar QoS to data while providing a better QoS to voice traffic.

We should however note that results for the DRMA protocol are only given for a low channel speed (i.e. 15 slots per 16 ms period) and the DR-TDMA protocol can not demonstrate its advantage at low speed since the control traffic is low and the FPBP

Table 4.5: Comparison between DR-TDMA and DRMA

Simulation Conditions	Number of Voice Stations at Voice Loss Rate of 1%	Data Throughput (packets/frame) at Voice Loss Rate of 1%
DRMA - $p_0 = 0.05$	26	1.2
DR-TDMA - $p_0 = 0.05$	26	1.1
DR-TDMA - $p_0 = 0.05$ - modified	26	1.3
DRMA - $p_0 = 0.2$	24	3.5
DR-TDMA - $p_0 = 0.2$	25	2.3
DR-TDMA - $p_0 = 0.2$ - modified	26	3.3

protocol is not useful for these conditions. The DRMA protocol uses fixed control packets transmission probabilities, therefore under a heavy load in high speed channels its performance will deteriorate while the DR-TDMA protocol can take advantage of the increasing statistical multiplexing gain to improve its performance and is not affected by the heavy contention traffic.

4.4.4 Results for Data, Voice and VBR Integrated System

In this section, we present simulation results to show that the integrated allocation algorithm respects the required QoS of each traffic category while providing an efficient utilization of the wireless channel. In order to completely investigate the performance of the integrated scheduling algorithm and the impact of each traffic and system parameter on the traffic QoS and protocol efficiency a huge amount of simulations are necessary. Since this investigation constitutes a research project on its own, we have restrained our simulations to a “proof of concept” of the DR-TDMA protocol.

Simulations have been run with 100 voice connections with parameters presented in Table 4.1, 100 data connections characterized by the parameters indicated in Table 4.6 and VBR connections with parameters showed in Table 4.7. The number of VBR con-

Table 4.6: Data connection parameters

Data source bit rate	19.2 Kbps
Average interarrival time between data burst (t_d)	100 ms
Beta parameter of the gamma function (β)	3
Mean burst length (μ_b)	5 cells
Maximum transfer delay (MTD)	60000 ms

Table 4.7: VBR connection parameters

Connection Bit Rate	250 Kbps
Basic source cell interarrival ($1/R_b$)	7.68 ms
Number of ON/OFF sources (S)	15
Average ON state length (t_b^{vbr})	100 ms
Average OFF state length (t_i^{vbr})	200 ms
Guaranteed parameter (g)	5
Guaranteed token pool depth (W)	1
Maximum transfer delay (MTD)	50 ms

nections is a varying parameter to illustrates the performance of the integrated system.

Figure 4.24 presents the voice and VBR (guaranteed and best effort traffic together) cell loss rates as a function of the number of VBR connections. The first thing that we can observe is that the integrated allocation algorithm meets the VBR and voice QoS (i.e. 1% cell loss rate) for all the traffic conditions. Even for a throughput of 99%, as indicated by Figure 4.25, the QoS of voice and VBR traffic are maintained at the expense of data QoS. We note that the cell loss rate of VBR traffic is higher than for voice traffic, this is due to the fact that a portion of the VBR traffic receives a best effort service. On the other hand, no VBR cells that conform to the guaranteed traffic parameters are lost (i.e. 0% VBR guaranteed cell loss rate). Even if VBR guaranteed and voice traffic receive the same QoS, the voice loss rate is higher than VBR guaranteed cell loss rate because of the voice cell losses that occur due to delayed control packets, which is not a determining factor for VBR guaranteed traffic performance.

The voice cell loss rate plateau that we observe between 13 and 14 VBR connections

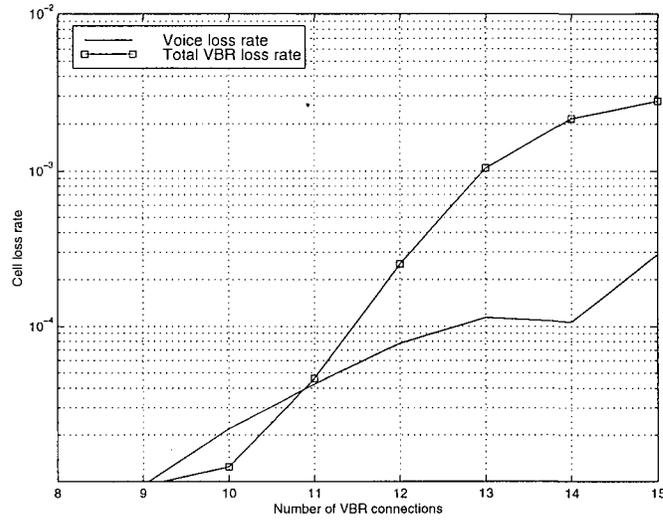


Figure 4.24: Cell loss rate as a function of the number of VBR connections for the integrated system.

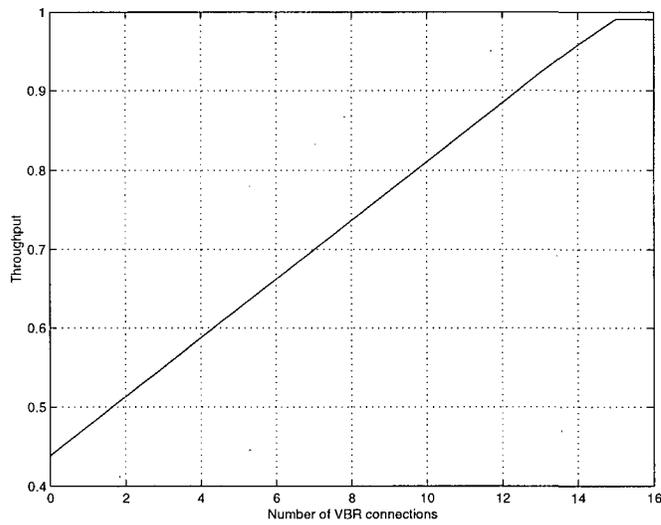


Figure 4.25: Throughput as a function of the number of VBR connections for the integrated system.

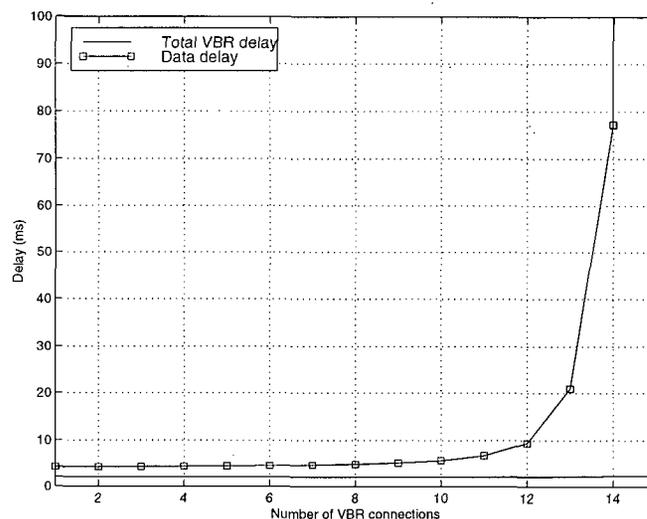


Figure 4.26: Delay as a function of the number of VBR connections for the integrated system.

might seem strange but similar results have been obtained for other traffic conditions. As explained before, the performance of voice traffic is mainly determined by contention traffic which, in its turn, is influenced by voice control traffic, data control traffic and number of control slots available. When the number of VBR connections increases, the number of available control slots decreases causing an increasing contention traffic which explains the increasing cell loss rate before the plateau. While, before the plateau the voice and control traffic is almost constant (the number of voice and data connections is constant), when we reach 13 VBR connections we see from the delay performance results in Figure 4.26 that the data delay dramatically increases. At this point the data control traffic decreases because more requests are sent to the base station piggybacked to data packets. Therefore, two effects (i.e. the decreasing number of control slots and the decreasing data control traffic) are opposed which creates the plateau. Then after, the decreasing number of available control slots is predominant which causes the increasing

voice cell loss rate.

Figure 4.26 shows that the VBR traffic (guaranteed and best effort traffic together) delay approximately remains at its minimum value of 2 ms for all throughput values. On the other hand, data traffic experiences a minimum delay of 4 ms (as explained in Section 4.3.2) while the throughput is below 80%. When the offered load increases, the allocation algorithm can maintain a reasonable data traffic delay below 100 ms up to a throughput of 96%. For higher offered loads, the data QoS rapidly decreases.

Chapter 5

Conclusion

To meet the expected increasing demand to connect mobile devices to the ATM broadband wired network we have proposed in this thesis a novel Dynamic Reservation TDMA MAC protocol for future wireless ATM networks. The DR-TDMA protocol efficiently integrates CBR, ABR and VBR traffic while maintaining their Quality-of-Service. The protocol features a novel framed pseudo-Bayesian priority algorithm that provides an improved uplink control management. The FPBP algorithm minimizes the contention delay and priority services can be provided. The DR-TDMA MAC protocol also optimizes the channel utilization by dynamically adjusting the uplink control period length as a function of the estimated control traffic. Furthermore, the DR-TDMA MAC protocol employs an efficient VBR rate-based allocation algorithm and a cell control algorithm used to determine cells that comply with the traffic parameters. This VBR allocation algorithm is integrated with CBR, voice and ABR allocation algorithms to efficiently share the wireless channel while maintaining the specific QoS requirements of each types of traffic.

5.1 Summary

We have investigated wireless ATM MAC protocol proposals using either CDMA or TDMA and we concluded that these protocols have the following problems:

- Inefficient protocol to manage the control access slots and to reduce the access delay of time-sensitive control packets;
- Inefficient allocation algorithm to integrate bandwidth sharing among control, CBR, ABR and VBR traffic.

To solve these problems we proposed the Dynamic Reservation TDMA (DR-TDMA) wireless ATM MAC protocol. We have adopted a reservation TDMA MAC protocol because it is efficient, flexible for slot allocation and it meets the requirements and constraints of a high speed wireless environment. Furthermore, most of the wireless ATM research projects use a reservation MAC protocol architecture. The DR-TDMA MAC protocol exhibits some similarities with the NEC MAC protocol but has enhanced features to solve the problems mentioned above. First, the DR-TDMA MAC protocol features the framed Pseudo-Bayesian priority access protocol that we have introduced in this thesis to manage the control slot access. Second, we propose a novel DR-TDMA allocation algorithm that efficiently integrates CBR, ABR, VBR and control traffic.

In this thesis, we have presented a new pseudo-Bayesian Aloha algorithm with priorities and we showed by our simulation results that it provides a significant delay improvement for high priority packet with both Poisson and self-similar traffic while low priority packets only experience slight performance degradation. The main advantages of this scheme over previously known priority protocols are its simplicity and its adaptation to the frame structure. Although this protocol is specifically designed for the WATM DR-TDMA MAC protocol, the slotted and framed priority protocols can be used in any situation where there is multiple traffic streams with different quality of service that are contending for the same channel. Our FPBP protocol is also well suited for any reservation MAC protocols where a certain amount of slots per frame are used for contention. The traffic in these slots can be well approximated as Poisson and consists of packets for

reservation at the beginning of a voice burst or data burst, requests for VBR bandwidth, hand-off requests, ... The FPBP protocol can be used to implement priorities among these traffic classes. Furthermore, the contention waiting time is an important factor in the overall performance of these MAC protocols. This hypothesis is confirmed by simulation results for the DR-TDMA protocol with data and voice integrated traffic. The performance results show that the QoS of voice traffic is significantly improved when the priority access protocol is used, while data traffic only suffers a slight QoS degradation.

We have also proposed new algorithms to allocate slots to voice, ABR, VBR and CBR traffic. For voice traffic (with voice activity detector) we have used a Time-To-Expiry service policy where packets for which their maximum waiting time will be exceeded are served first. The base station can predict, from the declared connection bit rates, the number of packets that are waiting in the mobiles. A similar approach is used for CBR traffic. For ABR traffic, the base station maintains the buffer length status of active connections through information about new packet arrivals in the mobiles sent either in control packets or piggybacked to data cells. Slots are allocated to ABR connections using a fair bandwidth allocation algorithm. We have further introduced a novel rate-based VBR cell scheduling algorithm where the centralized allocation algorithm predicts packet arrivals in mobiles by using information about the VBR cell arrival rates of the connections. Our scheduling strategy allocates slots based on the VBR connection's queue maintained in the base station and it provides a guaranteed service to cells that conform to the VBR traffic parameters and a best effort services for non-conforming cells. Guaranteed packets are served using a Time-To-Expiry approach while best effort cells receive slot allocations according to the fair bandwidth allocation algorithm. We have also integrated the new FPBP protocol to manage the access to uplink control slots and to determine the required control period length. Finally, we have proposed an algorithm that allows the system to integrate these ATM traffic while maintaining the required

QoS. The order of slot allocation priority is the following: CBR traffic, required control slots, voice and VBR guaranteed traffic, data (ABR) and VBR best effort traffic and, finally, the unused slots are also assigned for control traffic.

The simulation results show that the DR-TDMA performs at least as well as previously proposed protocol when only voice traffic is present. For a high speed channel, previous results indicate a performance degradation because of the contention in the control channel. However, the FPBP protocol minimizes the contention and over a high speed channel we can achieve a throughput of 96.5% while maintaining a voice cell loss rate below 1%. A theoretical analysis demonstrates that this is a near perfect performance.

For VBR traffic, the DR-TDMA protocol is efficient and can sustain a high throughput while maintaining low delays and cell loss rates for a wide variety of connections. Our protocol can offer a maximum throughput in the range of 90 to 95% over high speed channels and in the range of 85 to 90% over slower channels. This is much higher than what has been previously reported in the literature (in the range of 60 to 70% [29]).

When we integrate voice and data traffic, simulation results demonstrate that the FPBP protocol avoids a voice quality degradation under heavy traffic. Furthermore, for a high speed channel the DR-TDMA can sustain throughput in the range of 96.5 to 98% while maintaining a voice cell loss rate below 1% and data waiting delay under 250 ms. When the channel speed decreases, the throughput drops to around 83%. For integrated voice, data and VBR traffic, our simulation results show that the integrated allocation algorithm respects the required QoS of each traffic category (1% cell loss rate for VBR and voice traffic and data waiting delay under 250 ms) while providing an efficient wireless channel utilization of approximately 96%.

We have clearly demonstrated in this thesis that the proposed DR-TDMA MAC protocol is efficient to carry ATM traffic in a wireless ATM networks. It can offer QoS

guarantees to CBR, ABR and VBR traffic while providing a high channel throughput.

The performance of the proposed protocol can be explained by the following factors:

- The framed pseudo-Bayesian priority protocol minimizes the contention and reduces the access delay of time sensitive control packets;
- The dynamic nature of the uplink control period allows a high channel throughput when the control traffic is low and a small access delay when the control traffic increases;
- The efficient prediction algorithm for VBR traffic allows an efficient slot allocation for VBR connections with low waiting delays;
- The integrated allocation algorithm maximizes the channel throughput while meeting the required connection's QoS.

5.2 Topics for Future Investigations

In this thesis, we have described and analyzed the basic properties of the proposed new wireless ATM DR-TDMA MAC protocol. Further research remains to be done in order to validate and enhance this protocol for the future wireless ATM networks.

The VBR source model used in this thesis consists of a superposition of ON/OFF sources model. However, the presented results should be confirmed with real VBR streams from video coders such as the H.263 coder and MPEG4. Similarly, to validate results with data traffic, real TCP connections traffic can be used with the DR-TDMA MAC protocol. Furthermore, more thorough simulations for integrated traffic with diverse traffic scenarios should be run to investigate the behavior of the DR-TDMA MAC protocol under a wide range of conditions.

For simulations, we have considered an error-free wireless link. However, the wireless communication channel is error prone and the effect of channel errors on the behavior of the protocols should be studied. For example, the impact of control packets and VBR packets transmission errors on the performance of the DR-TDMA protocol should be carefully investigated. Enhancement to the DR-TDMA MAC protocol are also needed to facilitate an interaction with the Data Link Layer for packets retransmissions. For example, the priority to give to retransmitted packets and the allocation algorithm should be determined. Furthermore, some modifications will be needed in the VBR prediction algorithm to consider packet retransmissions.

In this thesis, we have only studied slot allocations for uplink traffic. However, these uplink allocation algorithms should be integrated with traditional scheduling algorithms for downlink slots in order to fairly allocate the frame slots to uplink and downlink connections.

For the pseudo-Bayesian priority algorithms, we have considered a perfect detection of idle, success and collision slots. Furthermore, we have not taken into account near-far effects and we have assumed that the exact transmission probability values can be transmitted to the mobiles (i.e. not the rounded values). The impact of these phenomena on the performance of the proposed pseudo-Bayesian priority protocols has to be carefully investigated.

Finally, an interesting area of research would be to integrate the wireless ATM link with a wired ATM network and to study the performance of the DR-TDMA MAC protocol in these conditions. The impact of the wireless link on the end-to-end performance of ATM connections can be therefore investigated. This will determine if the DR-TDMA MAC protocol is, as we believe, an essential part of an efficient wireless ATM network.

Bibliography

- [1] D. Raychaudhuri and N. D. Wilson, "ATM-Based transport architecture for multiservices wireless personal communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 1401–1414, October 1994.
- [2] D. Bertsekas and R. Gallager, *Data Networks*. Upper Saddle River, NJ: Prentice-Hall, second ed., 1992.
- [3] R. O. LaMaire, A. Krishna, P. Bhagwat, and J. Panian, "Wireless LANs and mobile networking: Standards and future directions," *IEEE Communications Magazine*, vol. 34, pp. 86–94, August 1996.
- [4] X. Qiu and V. O. Li, "Dynamic Reservation Multiple Access (DRMA): A new multiple access scheme for Personal Communications System (PCS)," *ACM/Baltzer Wireless Networks Journal*, vol. 2, pp. 117–128, June 1996.
- [5] Y. J. Kim, J. K. Kim, and Y. H. Lee, "A new medium access control scheme for wireless ATM networks," in *Proceedings VTC'97*, (Phoenix, AZ), May 1997.
- [6] O. Kubbar and H. T. Mouftah, "Multiple access control protocols for wireless ATM: Problems definition and design objectives," *IEEE Communications Magazine*, vol. 35, pp. 93–99, November 1997.
- [7] E. Ayanoglu, K. Y. Eng, and M. J. Karol, "Wireless ATM: Limits, challenges, and proposals," *IEEE Personal Communications*, vol. 3, pp. 18–34, August 1996.
- [8] D. Raychaudhuri, L. J. French, R. J. Siracusa, S. K. Biswas, R. Yuan, P. Narasimhan, and C. A. Johnston, "WATMnet: A prototype wireless ATM system for multimedia personal communication," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 83–95, January 1997.
- [9] N. Passas, S. Paskalis, D. Vali, and L. Merakos, "Quality-of-service-oriented medium access control for wireless ATM networks," *IEEE Communications Magazine*, vol. 35, pp. 42–50, November 1997.
- [10] K.-C. Chen, "Medium access control of wireless LANs for mobile computing," *IEEE Network*, vol. 8, pp. 50–63, September 1994.
- [11] D. Søbirk, J. M. Karlsson, L. Falk, and C. Lind, "An overview of proposed MAC algorithms for wireless ATM," in *The Thirteenth Nordic Teletraffic Seminar*, (NTNU, Trondheim, Norway), 1996. available at <http://www.tts.lth.se/Personal/daniels/papers/nts13.abstract.html>.

- [12] A. Acampora, "Wireless ATM: A perspective on issues and prospects," *IEEE Personal Communications*, vol. 3, pp. 8–17, August 1996.
- [13] Z. Liu, M. J. Karol, M. E. Zarki, and K. Y. Eng, "A demand-assignment access control for multi-code DS-CDMA wireless packet (ATM) networks," in *Proceedings of INFOCOM'96*, vol. 2, (San Francisco, CA), pp. 713–721, March 1996.
- [14] M. J. Karol, Z. Liu, and K. Y. Eng, "Distributed-queueing request update multiple access (DQRUMA) for wireless packet (ATM) networks," *Proceedings ICC'95*, pp. 1224–1231, June 1995.
- [15] G. Wu, H. Harada, K. Taira, and Y. Hase, "An integrated transmission protocol for broadband mobile multimedia communication systems," in *Proceedings VTC'97*, (Phoenix, AZ), May 1997.
- [16] S. G. Fischer, T. A. Wysocki, and H.-J. Zepernick, "MAC protocol for a CDMA based wireless ATM LAN," in *Proceedings ICC'97*, (Montréal, Canada), June 1997.
- [17] S. Dastangoo, "A multimedia access control protocol for ATM based mobile networks," in *Proceedings PIMRC'95*, vol. 2, (Toronto, Canada), pp. 794–798, September 1995.
- [18] R. Pichna and Q. Wang, "A MAC protocol for the integrated wireless access network," in *Proceedings PIMRC'95*, vol. 1, (New-York, NY), pp. 248–252, September 1995.
- [19] A. S. Mahmoud, D. D. Falconer, and S. A. Mahmoud, "A multiple access scheme for wireless access to a broadband ATM LAN based on polling and sectored antennas," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 596–608, May 1996.
- [20] D. Falconer, "A system architecture for broadband millimeter-wave access to an ATM LAN," *IEEE Personal Communications*, vol. 3, pp. 36–41, August 1996.
- [21] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Transactions on Communications*, vol. 37, pp. 885–890, August 1989.
- [22] R. Bolla, F. Davoli, M. Taffone, and F. Reichert, "The PRMA-ISA protocol for multiple access of mixed voice and data traffic," in *Proceedings VTC'94*, vol. 2, (Stockholm, Sweden), pp. 1213–1217, June 1994.
- [23] M. Thomas and D. J. Goodman, "Multi-rate PRMA: A time division protocol for adjustable bit-rate sources," in *Proceedings VTC'97*, (Phoenix, AZ), May 1997.

- [24] G. Bianchi, F. Borgonovo, L. Fratta, L. Musumeci, and M. Zorzi, "C-PRMA: The centralized packet reservation multiple access for local wireless communications," in *Proceedings GLOBECOM'94*, (San Francisco, CA), pp. 1340–1345, November 1994.
- [25] J. Dunlop, J. Irvine, D. Robertson, and P. Cosimini, "Performance of a statistically multiplexed access mechanism for a TDMA radio interface," *IEEE Personal Communications*, vol. 2, pp. 56–64, June 1995.
- [26] N. Amitay and S. Nanda, "Resource auction multiple access (RAMA) for statistical multiplexing of speech in wireless PCS," *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 584–596, August 1994.
- [27] D. Raychaudhuri, "Wireless ATM networks: Architecture, system design and prototyping," *IEEE Personal Communications*, vol. 3, pp. 42–49, August 1996.
- [28] H. Xie, P. Narasimhan, R. Yuan, and D. Raychaudhuri, "Data link control protocols for wireless ATM access channels," in *Proceedings ICUPC'95*, (Tokyo, Japan), pp. 753–757, November 1995.
- [29] S. K. Biswas, D. Reininger, and D. Raychaudhuri, "UPC base bandwidth allocation for VBR video in wireless ATM," in *Proceedings ICC'97*, (Montréal, Canada), June 1997.
- [30] J. G. Kim and I. Widjaja, "PRMA/DA: A new media access control for wireless ATM," in *Proceedings of ICC/SUPERCOMM'96*, vol. 1, (Dallas, TX), pp. 240–244, June 1996.
- [31] B. Walke, D. Petras, and D. Plassmann, "Wireless ATM: Air interface and network protocols of the mobile broadband system," *IEEE Personal Communications*, vol. 3, pp. 50–56, August 1996.
- [32] H. Morikawa, "Wireless access for adaptive video applications," in *Proceedings VTC'97*, (Phoenix, AZ), May 1997.
- [33] X. Wu, S. Wu, H. Sun, and L. Li, "Dynamic slot allocation multiple access protocol for wireless ATM networks," in *Proceedings ICC'97*, (Montréal, Canada), June 1997.
- [34] W. C. Chan and E. Geraniotis, "A medium access protocol for interconnecting ATM and wireless networks," in *Proceedings ICC'97*, (Montréal, Canada), June 1997.
- [35] D. Petras, "Medium access protocol for wireless, transparent access," in *Proceedings Wireless Communication Systems Symposium*, (Long Island, NY), November 1995. available at <http://www.comnets.rwth-aachen.de/~petras/Publications.html>.

- [36] D. Petras and A. Krämling, "MAC protocol with polling and fast collision resolution for an ATM air interface," in *Proceedings IEEE ATM Workshop*, (San Francisco, CA), August 1996. available at <http://www.comnets.rwth-aachen.de/~petras/Publications.html>.
- [37] S. Choi and K. G. Shin, "A cellular wireless local area network with QoS guarantees for heterogeneous traffic," in *Proceedings INFOCOM'97*, (Kobe, Japan), April 1997.
- [38] P. Agrawal, E. Hyden, P. Kryzanowski, P. Mishra, M. B. Srivastava, and J. A. Trotter, "SWAN: A mobile multimedia wireless network," *IEEE Personal Communications*, vol. 3, pp. 18–33, April 1996.
- [39] M. B. Srivastava, "Medium access control and air-interface subsystem for an indoor wireless ATM network," in *Proceedings of 9th International Conference on VLSI Design '96*, (Bangalore, India), pp. 14–18, January 1996.
- [40] N. D. Wilson, R. Ganesh, K. Joseph, and D. Raychaudhuri, "CDMA versus dynamic TDMA for access control in an integrated voice/data PCN," in *Proceedings ICUPC'92*, (Dallas, TX), pp. 267–272, September 1992.
- [41] N. D. Wilson, R. Ganesh, K. Joseph, and D. Raychaudhuri, "Packet CDMA versus dynamic TDMA for multiple access in an integrated voice/data PCN," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 870–884, August 1993.
- [42] J. G. Lee and M. S. Corson, "The performance of an "imbedded" Aloha protocol in wireless networks," in *Proceedings of PIMRC'96*, vol. 2, (Taipei, Taiwan), pp. 377–381, October 1996.
- [43] P. Papantoni-Kazakos, "Multiple-access algorithms for a system with mixed traffic: High and low priority," *IEEE Transactions on Communications*, vol. 40, pp. 541–555, March 1992.
- [44] W. S. Chung and C. K. Un, "Collision resolution algorithm for M -priority users," *IEE Proceedings-Communications*, vol. 142, pp. 151–157, June 1995.
- [45] D. Makrakis, R. S. Mander, and P. Papantoni-Kazakos, "A spread slotted medium access control protocol with reservation supporting multimedia services over personal and mobile communication networks," in *Proceedings of ICC/SUPERCOMM'96*, vol. 2, (Dallas, TX), pp. 813–817, June 1996.
- [46] M. Paterakis and P. Papantoni-Kazakos, "A simple window random access algorithm with advantageous properties," *IEEE Transactions on Information Theory*, vol. 35, pp. 1124–1130, September 1989.

- [47] D. Makrakis, R. S. Mander, and L. Orozco-Barbosa, "A spread slotted random access control protocol with multi-priority for personal and mobile communication networks carrying integrated traffic," in *Proceedings of 1995 Canadian Conference on Electrical and Computer Engineering*, vol. 1, (Montréal, Canada), pp. 132–136, September 1995.
- [48] I. Stavrakakis and D. Kazakos, "A multiuser random-access communication system for users with different priorities," *IEEE Transactions on Communications*, vol. 39, pp. 1538–1541, November 1991.
- [49] T. Papantoni-Kazakos, N. B. Likhanov, and B. S. Tsybakov, "A protocol for random multiple access of packets with mixed priorities in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1324–1331, September 1995.
- [50] M. Liu and P. Papantoni-Kazakos, "A random access algorithm for data networks carrying high priority traffic," in *Proceedings of INFOCOM'90*, vol. 3, (San Francisco, CA), pp. 1087–1094, June 1990.
- [51] M. Paterakis and Y. Gong, "Performance analysis of random access multiuser algorithms for packets with different priorities," in *Proceedings of INFOCOM'90*, vol. 2, (San Francisco, CA), pp. 580–587, June 1990.
- [52] R. L. Rivest, "Network control by Bayesian broadcast," *IEEE Transactions on Information Theory*, vol. 33, pp. 323–328, May 1987.
- [53] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, third ed., 1991.
- [54] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1–15, February 1994.
- [55] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, June 1995.
- [56] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proceedings of ACM Sigcomm'94*, vol. 24, (London, UK), pp. 269–280, September 1994.
- [57] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in Variable-Bit-Rate video traffic," *IEEE Transactions on Communications*, vol. 43, pp. 1566–1579, February/March/April 1995.
- [58] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, second ed., 1994.

- [59] D. Petras and A. K. A. Hettich, "MAC protocol for wireless ATM: Contention free versus contention based transmission of reservation requests," in *Proceedings PIMRC'96*, (Taipei, Taiwan), October 1996. available at <http://www.comnets.rwth-aachen.de/~petras/Publications.html>.
- [60] J. Y. Hui, "Predictive queuing multiple access - A wireless ATM protocol for multimedia communication," in *Proceedings VTC'97*, (Phoenix, AZ), May 1997.
- [61] The ATM Forum, "The ATM Forum traffic management specification." v. 4.0, 1996.
- [62] R. Jain, S. Kalyanaraman, S. Fahmy, R. Goyal, and S.-C. Kim, "Source behavior for ATM ABR traffic management: An explanation," *IEEE Communications Magazine*, vol. 34, pp. 50-57, November 1996.
- [63] R. Jain, "Congestion control and traffic management in ATM networks: Recent advances and a survey," *Computer Networks and ISDN Systems*, vol. 28, pp. 1723-1738, 1996.
- [64] A. Arulambalam, X. Chen, and N. Ansari, "Allocating fair rates for available bit rate service in ATM networks," *IEEE Communications Magazine*, vol. 34, pp. 92-100, November 1996.
- [65] S. Nanda, D. J. Goodman, and U. Timor, "Performance of PRMA: A packet voice protocol for cellular systems," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 584-598, August 1991.
- [66] H. Heeke, "A traffic-control algorithm for ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, June 1993.
- [67] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Transactions on Circuit and Systems for Video Technology*, October 1998.
- [68] K. Joseph and D. Reininger, "Source traffic smoothing and atm network interfaces for vbr mpeg encoders," in *Proceedings of ICC'95*, vol. 3, (Seattle, WA), pp. 1761-1767, June 1995.
- [69] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Transactions on Communications*, vol. 36, pp. 834-844, July 1988.
- [70] V. Paxson, "Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic," *ACM Sigcomm*, vol. 27, pp. 5-18, October 1997.

- [71] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second ed., 1992.

Appendix A

Generation of Self-Similar Traffic

For our needs, we will consider the number of packets arrival per slot to be the self-similar stochastic process $\{X_t\}_{t=0,1,\dots}$ to generate. This is a discrete-time, discrete-value process. We have not found in the literature any method to generate a stochastic process of this type with strictly specified statistics. Instead, we will generate an *asymptotically self-similar* stochastic process.

Following [70], a stochastic process $\{X_t\}_{t=0,1,\dots}$ with auto-correlation $r(k)$ is said to be *asymptotically self-similar* if:

$$r(k) \sim k^{(-2-2H)} L(k), \quad \text{as } k \rightarrow \infty \quad (\text{A.1})$$

where the Hurst Parameter H satisfies $1/2 < H < 1$ and L is a slowly varying function, that is, $\lim_{t \rightarrow \infty} L(tx)/L(t) = 1$, for all $x > 0$. We thus have $\sum_k r(k) = \infty$ which illustrates the long-range dependency of such a process. For a more thorough discussion on long-range dependency and self-similar stochastic process definitions and properties see [54, 57].

It has been shown that the buffer occupancy process in an $M/G/\infty$ queue, where customers arrive according to a Poisson process and have *heavy-tailed* distributed service time with infinite variance, results in an asymptotically self-similar count process [54]. This model implies that multiplexing constant-rate connections that have a Poisson arrival process and a heavy-tailed connection lifetime distribution with infinite variance will result in an asymptotically self-similar data traffic [55].

A distribution is defined as heavy-tailed if:

$$P[X \geq x] \sim cx^{-\beta}, \quad \text{as } x \rightarrow \infty, \beta \geq 0 \quad (\text{A.2})$$

The Pareto distribution is an example of a heavy-tailed distribution [55]. However, this distribution is continuous and we need a discrete heavy-tailed distribution since the duration of a service in our case is in number of packets (which is discrete). We thus need to introduce a discrete heavy-tailed distribution with infinite variance but finite mean. We propose the distribution with the following probability density function:

$$p[X = x] = 4/(x(x+1)(x+2)) \quad \text{for } x \geq 1 \quad (\text{A.3})$$

We can show that this distribution has the following statistics:

$$p[X \geq x] = 2/(x(x+1)) \sim cx^{-2}, \quad \text{as } x \rightarrow \infty \quad (\text{A.4})$$

$$E[X] = 2 \quad (\text{A.5})$$

$$\text{Var}[X] = \sum_{x=1}^{\infty} \frac{4x}{(x+1)(x+2)} = \infty \quad (\text{A.6})$$

Connection lifetime in number of packets can thus be generated according to the proposed distribution since it satisfies the heavy-tailed with infinite variance property. This distribution can be generated with the rejection method [71] using the following comparison function:

$$f(x) = \begin{cases} 2/3 & \text{if } 1 \leq x \leq 2, \\ 5/x^3 & \text{if } x > 2. \end{cases} \quad (\text{A.7})$$

Appendix B

List of Abbreviations and Acronyms

ABR	Available Bit Rate
AE	Allocation Efficiency
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BT	Burst Tolerance
CBR	Constant Bit Rate
CDF	Cumulative Density Function
CDMA	Code Division Multiple Access
CSMA/CA	CSMA with collision avoidance
CSMA/CD	CSMA with collision detection
CSMA	Carrier Sense Multiple Access
D-TDMA	Dynamic-TDMA
DQRUMA	Distributed-Queuing Request Update Multiple Access
DR-TDMA	Dynamic Reservation TDMA

DRMA	Dynamic Reservation Multiple Access
DS-CDMA	Direct-Sequence CDMA
DSA++	Dynamic Slot Assignment
DSAMA	Dynamic Slot Allocation Multiple Access
DSMA	Dynamic Slot Multiple Access
FAPBP	Framed Adaptation Pseudo-Bayesian Priority
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FIFO	First In First Out
FPBP	Framed Pseudo-Bayesian Priority
GRAP	Group Randomly Addressed Polling
MAC	Medium Access Control
MBS	Maximum Burst Size
MC-CDMA	Multi-Code CDMA
MDR-TDMA	Multiservice Dynamic Reservation TDMA
MTD	Maximum Transfer Delay
NC-PRMA	Non-collision PRMA
NRTT	Non-Real Time Traffic

PCR	Peak Cell Rate
PRMA/DA	PRMA with Dynamic Allocation
PRMA	Packet Reservation Multiple Access
PRMA-ISA	PRMA-Independent Stations Algorithm
QoS	Quality of Service
RAMA	Resource Auction Multiple Access
RTT	Real Time Traffic
SPBP	Slotted Pseudo-Bayesian Priority
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TDM	Time-Division Multiplexing
TTE	Time-To-Expiry
UBR	Unspecified Bit Rate
VBR	Variable Bit Rate
WATM	wireless ATM
WLAN	Wireless Local Area Network