QUALITY–OF–SERVICE BASED BANDWIDTH ALLOCATION IN INTEGRATED MULTISERVICES WIRELESS NETWORKS

BY PARTHASARATHY NARASIMHAN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Operations Research

Written under the direction of

Roy D. Yates

and approved by

New Brunswick, New Jersey

May, 1996

ABSTRACT OF THE DISSERTATION

Quality–of–Service based bandwidth allocation in integrated multiservices wireless networks

by Parthasarathy Narasimhan, Ph.D.

Dissertation Director: Roy D. Yates

Third–generation wireless information networks will be required to support integrated multimedia services, with high penetration and high throughput. The demand for these services is expected to increase greatly. With high user densities, effective usage of the bandwidth at the medium access layer (MAC) is necessary to realize the expected throughput, while maintaining quality–of–service (QoS) requirements of user applications.

In this dissertation, we study bandwidth allocation policies at the MAC layer of integrated, multiservices wireless networks. In particular, we study two time–slotted systems – Packet Reservation Multiple Access (PRMA) and Wireless ATM – and propose bandwidth allocation policies that meet the QoS requirements of user applications.

We propose a variant of PRMA, called Frame Reservation Multiple Access (FRMA), to reduce receiver activity, extend battery life, and to accommodate time–division duplex (TDD) of uplink and downlink transmissions. We present an exact analysis of an FRMA system by modeling it as a Markov Chain. We formulate an optimization problem to choose the design parameters with a view to optimizing system performance.

We extend the voice–only FRMA protocol to integrate voice and data sources in the same system using a dynamic movable boundary scheme. Bandwidth for the data

subsystem is allocated without affecting the capacity of the system to support voice sources. To choose design parameters to maximize expected bandwidth available for the data subsystem, we formulate an optimization problem and obtain heuristic solutions. Even though the solutions to the above problem are approximate, the computed performance measures by implementing these solutions are accurate.

A description of a wireless ATM network followed by a discussion of the issues relevant to the design of the Medium Access Control layer and the Data Link Control (DLC) layer follows. The base station in the wireless ATM network has complete control over bandwidth allocation to the many users in the cell. We propose a solution to the problem of bandwidth allocation for variable bit–rate (VBR) sources in this system with centralized bandwidth allocation. We propose two criteria that could be used to distribute the inevitable packet loss amongst the users in such a way that each user gets an *even* share of the bandwidth. We conjecture that using either of the two criteria mentioned above to allocate bandwidth for VBR sources leads to the convergence QoS experienced by each source in the system to a constant value. Assuming these conjectures were true, we derive a simple expression that would determine the feasibility of the QoS requirements of the sources given the bandwidth that is available. This expression could be used for connection admission control (CAC) purposes.

Acknowledgements

I wish to thank my advisor, Dr. Roy Yates, for his able guidance throughout the course of my dissertation work. I have been indeed fortunate to have had the pleasure of working with him for a period of more than three years. In spite of his busy schedule and commitments, he has always found time to discuss my work and provide effective feedback. I have always looked forward to our weekly meetings where we have discussed a wide range of interesting topics. I will miss these meetings very much.

I thank Dr. David Goodman for providing me an opportunity to be a part of WINLAB and offering to support my research work. His long years of experience in the field and his unique perceptive ability to determine what is good have resulted in timely advice towards my choosing a good dissertation topic. Dr. Jack Holtzman's comments and suggestions at various stages of this dissertation have added value to this work. I have also enjoyed interacting with everyone at WINLAB.

I have also had the excellent opportunity of being associated with C & C Research Labs, NEC USA. I thank Dr. D. Raychaudhuri for it and also for his guidance and direction during the later part of my dissertation work. I have benefited greatly from my participation in the WATMnet prototype development project. My interaction with the members of this premier research organization, Dr. Bhaskar Sengupta, Dr. G. Ramamurthy, and Dr. Arif Merchant (now with HP Labs) in particular, have contributed significantly to my professional development.

At a time when I was in search of a dissertation topic, Dr. Mike Rothkopf suggested that I include WINLAB as one of my options. In retrospect, it looks like a golden piece of advice for I have immensely enjoyed my work at WINLAB. I thank Dr. Rothkopf for also being a member of my dissertation committee and providing useful feedback on my thesis. His course, "Case studies in applied OR", is unique in its content and organization, and is most enjoyable.

I would also like to thank Dr. Endre Boros and Dr. Andras Prekopa for participating in my dissertation committee at very short notice. I have benefited greatly from my association with RUTCOR. I thank Dr. Peter Hammer for providing me with this opportunity and support.

I have never had to concern myself with a lot of administrative issues at Rutgers. All the credit for it has to go to the excellent department secretaries – Clare Smietana and Terry Hart at RUTCOR, and Phil Genatempo at WINLAB. They insulate the students from mundane administrative issues, thus permitting the students to concentrate more on their academic work. Clare's help, especially with the paperwork before my defense, while I was in Dallas, deserves special mention. Phil's remarkable ability to remain cheerful under all circumstances is greatly appreciated.

Last, but not the least, I would like to thank my wife, Vasu (Priya to others), for providing constant support throughout my dissertation work. She has done more than her fair share of the household chores (they increased by an order of magnitude with the birth of our daughter, Maya) so I could concentrate on my research. She has provided gentle reminders of what my goals are whenever I strayed from the path towards these goals. I hope I will be able to reciprocate when she begins her journey towards her PhD dissertation now.

Dedication

To my parents

whose dreams and sacrifices have carried me to this goal

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AMPS Advanced Mobile Phone System

ATM Asynchronous Transfer Mode

B-ISDN Broadband Integrated Services Digital Network

CDMA Code Division Multiple Access

CSMA Carrier Sense Multiple Access

DLC Data Link Control

FDD Frequency Division Duplex

FDMA Frequency Division Multiple Access

FRMA Frame Reservation Multiple Access

GSM Global System for Mobile Communications

ISDN Integrated Services Digital Network

MAC Medium Access Control

PRMA Packet Reservation Multiple Access

QoS Quality of Service

TDD Time Division Duplex

TDMA Time Division Multiple Access

VC Virtual Circuit

Chapter 1

Introduction

There seems to be a major trend in person–to–person communications in moving from wired fixed place–to–place communications to a wireless, mobile communications. The existence of this trend has been enabled by the rapid growth of mobile technologies that provide primarily two–way voice services even though economical wireline services[1] are readily available. The next generation of wireless information networks, referred to as personal communication networks (PCN)[2], are expected to provide services in addition to the two–way voice services of today's mobile cellular systems. Some of these possible services/applications include data and messaging services, video teleconferencing and other video services, and other forms of multimedia services. There is a lot of research and development activity to integrate today's diverse networks into one network that would provide most of these services. These integrated networks would collect information from diverse sources and deliver them to their destinations. The next generation of PCNs, will be required to provide integrated multimedia services with high penetration and high throughput [10].

There is widespread anticipation that consumer demand for wireless personal communication services will continue to expand for the foreseeable future. The high capacity required to support these services can be achieved by using microcells with cell sizes as small as possible and channel reuse after a short distance. Digital cellular and microcellular radio systems must incorporate multiple access schemes that make efficient use of the allocated bandwidth and radio cell infrastructure with minimum cost and maximum performance, which is necessary for achieving the high throughput that is required by

[1]In fact, the quality of the mobile services are, even today, inferior compared to the wireline services.

[2]Referred to also as Personal Communication Services (PCS)

the applications to be supported by the next generation of personal communication networks.

It has become clear that the traditional circuit–switching approach, where bandwidth is reserved exclusively for the use of one connection is not efficient for current applications. Statistical multiplexing of many connections on a single link to exploit the variations of the bit–rate produced by the information source can lead to higher utilization of the available bandwidth over the traditional methods. But statistical multiplexing leads to *blackout periods*, periods when the channel may not be available to certain sources even though they may have information to transmit. Thus, stringent quality of service (QoS) requirements have to be met while admitting sources into the system and also in distributing available bandwidth among those competing sources that have been already admitted into the system. In this dissertation, we concentrate on the latter part, *i.e.* once the sources are admitted into the system, the problem of bandwidth allocation subject to satisfying QoS requirements will be addressed here.

The input to an integrated wireless PCN will be a combination of sources that generate information with vastly different characteristics and QoS requirements. For example, voice and video applications can tolerate some information loss, but are very sensitive to delay, i.e. voice/video information cannot be delayed beyond a certain delay limit and have to be transmitted within this limit. On the other hand, data applications (file transfer, for example) can tolerate some delay, but no information loss. Thus the QoS requirements of each source are decided by the nature of information that the source generates. This study concentrates on the performance achieved by the delay–sensitive sources where the QoS parameter of interest is information loss probability.

In this dissertation, we propose to discuss bandwidth allocation schemes at the Medium Access Control (MAC) layer of wireless networks. Given a statistical description of the source characteristics, and the QoS parameters required by the sources, the goal is to design bandwidth allocation policies so as to maximize utilization while maintaining the QoS parameters within acceptable limits. We also attempt to use as little bandwidth as possible in achieving the QoS objectives. The reason for this is that the sources under consideration here are those that have delay–sensitive information to

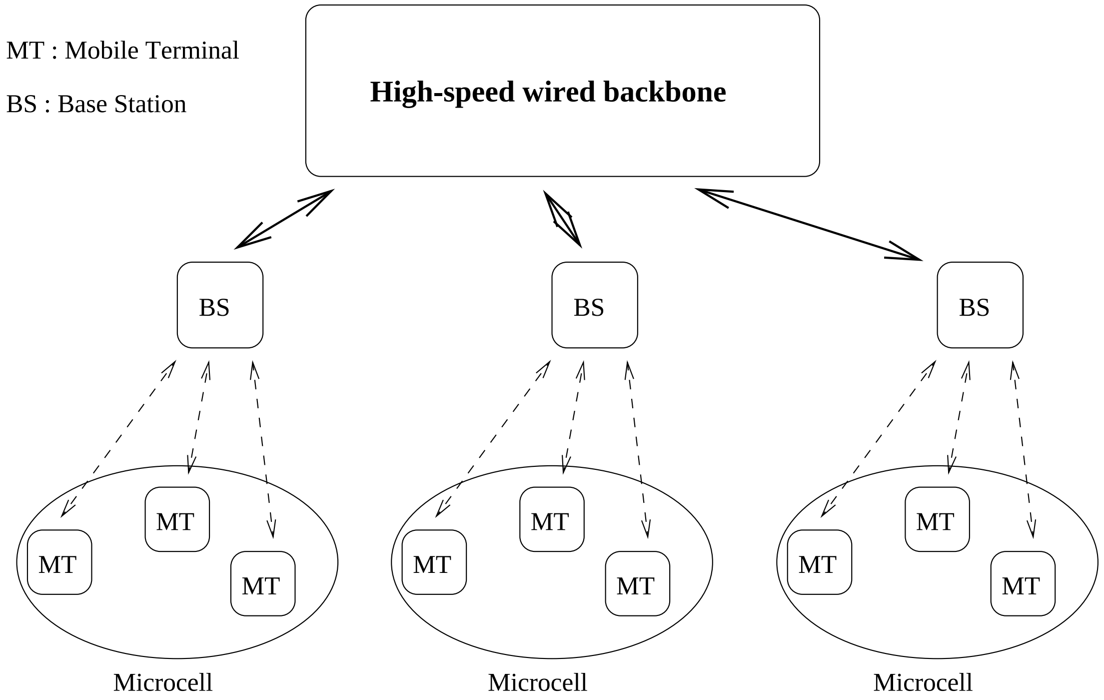


Figure 1.1: Hierarchy of wireless network

send. We assume that there will always be sources that will be served on a *best–effort* mode, *i.e.* any bandwidth that is left over after the delay–sensitive sources have been served will be used to serve these delay–insensitive sources. Thus, the goal of minimizing bandwidth use for the delay–sensitive sources is justified because it maximizes the bandwidth available for the *best–effort* applications.

1.1 Wireless network architecture

Figure 1.1 shows a typical hierarchical wireless network. The base station provides coverage to mobile stations in its cell. The base stations are assumed to be connected to a high–speed wireline backbone network. This dissertation focuses on the efficient allocation of bandwidth over the air–interface. No assumptions are made about the architecture of the wireline backbone network. Some of the possibilities for the backbone network are a Cellular Packet Switch, as described in [35], or a standard ATM network [54].

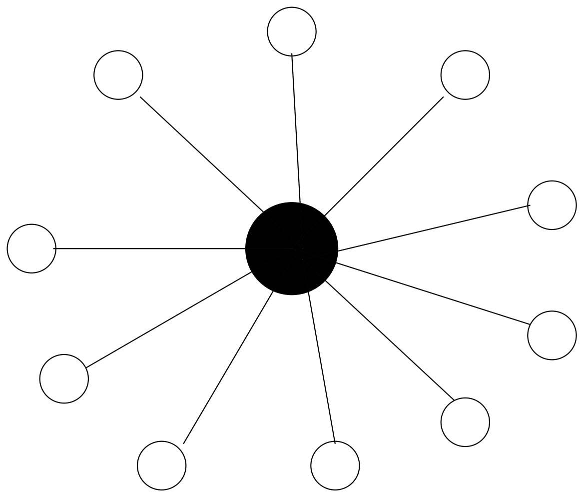


Figure 1.2: Star topology

The topology for the air–interface will be a star network, *i.e.* a central base station that is *connected* to each of the mobile terminals with a wireless link (Figure 1.2). The mobile terminals share a common bandwidth to communicate with the base station. The base station, on the other hand, broadcasts its information and all the mobile terminals are assumed to be able to receive this broadcast.

The bandwidth allocation problem in two different systems will be addressed in this dissertation. These two systems are based on Time Division Multiple Access (TDMA), which is one of the many access schemes in which several users share the available bandwidth. In TDMA, the users get access to the channel on a time–sharing basis, where the entire bandwidth is assigned to a particular user for a short period of time, called a time slot. This assignment is repeated with a certain periodicity and the time interval over which it is repeated is usually called a frame. A more detailed description of TDMA, along with the other multiaccess schemes, can be found in Chapter 2.

1.2 Packet Reservation Multiple Access

The first system that this dissertation considers is Packet Reservation Multiple Access (PRMA) [19], a multiaccess protocol that increases the capacity of TDMA by exploiting the talkspurts and silence periods in conversational speech and allocating bandwidth to speech terminals only during talkspurts. At the end of every slot, the base station broadcasts a feedback message that contains information on the state of that time slot in the next frame. Each time slot is classified as either reserved by a voice terminal or as available. Those that are reserved by a certain voice terminal are meant for the exclusive use of that particular terminal until the terminal completes its talkspurt and gives up the use of this time slot. At this point, the time slot is classified as available in future frames. All voice terminals, currently in a talkspurt, but without a reserved slot, can contend for available slots by transmitting the oldest packet in their buffer. If exactly one terminal contends in a slot, then this contention is termed successful and the terminal obtains a reservation for this time slot. A more detailed discussion can be found in Chapter 3.

Speech packets have a limited lifetime, *i.e.* they lose their significance if they are delayed beyond a certain limit. In PRMA, speech packets may be delayed for one of two reasons :

- A speech terminal begins a talkspurt, but is unable to contend for a channel because there are no available time slots.
- There are time slots available for contention, but the speech terminal has not been successful in its contention.

Delayed speech packets are dropped. Since dropped speech packets affect speech quality, packet dropping probability is an important measure of PRMA performance.

At the end of every time slot, the base station broadcasts an acknowledgment message that indicates whether the base station was able to decode the packet it received during that slot. If the base station successfully decodes a packet, it identifies the terminal that transmitted the packet. Otherwise, it transmits a NULL acknowledgment. A NULL acknowledgment message could be a result of either no transmission or transmission by more than one terminal at the same time. Since the mobile terminals get the system state information from this feedback message, they are required to listen in every slot. This results in high receiver activity at the mobile terminal and hence reduces battery life. Also, the base station is expected to process all the information it received in a time slot and broadcast a feedback at the end of the slot.

In order to reduce receiver activity at the mobile terminal and extend battery life, we have suggested modifications to PRMA. In the resultant protocol, called Frame Reservation Multiple Access (FRMA), the base station broadcasts the acknowledgments for each slot in the frame at the end of the frame. In addition to preventing the mobile stations needing to have their receivers switched on in every time slot, this protocol also enables time–division–duplexing (TDD) of uplink and downlink frames. Processing delays at the base station can be accounted for by scheduling the acknowledgments appropriately in the downlink frame.

An exact analysis of the performance of a voice–only FRMA has been carried out by constructing a Markov chain model of the system. We present four different approaches for choosing design parameters for the system appropriately. We formulate a non–linear programming problem to choose design parameters in order to minimize voice packet dropping probability and use a direct–search technique to find approximate solutions to the above problem. For a particular set of system parameters, using any of the above techniques to choose the design parameters, the FRMA system had a slightly higher capacity than a PRMA system with identical system parameters.

1.3 Voice-Data integration in FRMA

The systems based on PRMA and FRMA, described in the previous section, were assumed to carry voice traffic only. Previous studies by Nanda [38] and Wong and Goodman [62] have attempted to integrate voice and data traffic by letting the data terminals contend for the available slots along with the contending voice terminals. We propose a new protocol for the integration of voice and data in FRMA. Unlike the abovementioned studies on voice–data integration in PRMA [38, 62], we propose a logical separation of the voice and the data subsystems, with a movable boundary between the two subsystems to ensure the Quality–of–Service (QoS) requirements of the voice subsystem are met, while maximizing the expected available bandwidth for the data subsystem.

The available bandwidth is divided into three regions – the voice region, the voice contention region, and the data region. All voice terminals with reservations are assigned a time slot in the voice region. The size of this region grows and shrinks according to the arrival[3] and departure of talkspurts. The voice contention region, along with the superposition of the speech processes from all the voice terminals, controls the rate at which contending voice terminals obtain reservations and move into the voice region. The size of the voice contention region is controlled so as to meet the QoS requirements of the voice users. The data region is for the use of the data terminals. The size of the data region depends heavily on the current state of the voice terminals and the size of the voice contention region.

We formulate an optimization problem to choose the design parameters appropriately in order to maximize the available bandwidth for the data users while maintaining the QoS requirements of the voice users and propose a heuristic approach to solve this problem. The results, using a particular set of system parameters, indicate that a significant amount of bandwidth can be freed for the data subsystem. We studied the performance of the data subsystem by embedding a simple data protocol in the data region. The study using simulations shows that that the sustainable offered load with this system is significantly higher than that of the integrated voice–data PRMA systems described in the literature [38, 62].

1.4 Centralized Medium Access Control

In PRMA, the base station provides feedback on whether it was able to decode the packet it received in a time slot. The base station is not required to play an active role in assigning time slots to the mobile terminals. The mobile terminals contend for available slots and depending on the results of the contention, they are assigned time slots for their use till the end of their talkspurts. Also, the base station has no knowledge of the start of the talkspurts at the mobile stations and the duration of these talkspurts. Hence, the base station has limited knowledge of the dynamic bandwidth

[3]The arrival is into the reserved state, *i.e.* when a voice terminal contends successfully and obtains a reservation.

requirements of the mobile stations.

The next part of the dissertation will focus on a system in which the base station plays an active role in the allocation of bandwidth to the mobile terminals. We assume the base station to have periodic information on the dynamic bandwidth requirements (the bandwidth requirements are assumed to be variable with time) of the information source at mobile stations[4]. We also assume that the available bandwidth is less than the sum of the peak requirements of the mobile stations[5]. The mobile stations inform the base station of their current bandwidth requirements periodically. Given that there are going to be periods when all of the requirements cannot be met, the base station has to partition the available bandwidth in such a manner that each mobile terminal gets a *even* share of the bandwidth.

The sum of the peak requirements of the sources admitted into the system is assumed to be higher than the available bandwidth. Hence, some packet loss is inevitable. We assume that the packets offered to the system in each time unit expire at the end of that time unit. Thus, any packet that cannot be accepted by the base station for transmission in that time unit is lost/dropped. The Quality–of–Service (QoS) parameter of interest here is packet loss. At the time of connection set up, the mobile terminals inform the base station of target (or acceptable) QoS that they would like to achieve. That is, the mobile terminals inform the base station of what their maximum packet loss should be. If the achieved packet loss for a given source is less than the target specified, then the QoS criterion is said to be satisfied for this particular source.

The channel in this system, as in PRMA, is assumed to be time–slotted and the duration of a slot is assumed to be long enough to transmit one packet. Time slots are grouped together into frames, which will be the time unit over which the sources at the mobile terminals inform the base station of their bandwidth requirements. The bandwidth is allocated in integral number of time slots in each frame. The base station keeps an account, from the start of the connection to the present frame, of the number

[4]The terms "source" and "mobile terminal" will be used interchangeably to mean the same entity.

[5]Otherwise, the bandwidth allocation problem has a simple solution – allocate whatever the mobile terminal requires.

of packets offered to the system by each source and the number of packets that have been dropped from each source. The ratio of these two quantities is the current packet loss faced by each source. This ratio is normalized by the target QoS specified by each source. We will call this quantity the normalized packet loss.

We propose two criteria to decide on how to distribute this inevitable packet loss (if any) among the sources in the system in each time unit. The first criterion, called the *minmax criterion,* is based on minimizing the maximum normalized packet loss faced by any source. At the start of any frame in which all of the packets offered to the system cannot be carried, the base station tries to distribute the packet loss among all the sources in such a way that at the end of the frame, the maximum normalized packet loss faced by any source is minimum. The second criterion, called the *maxmin criterion,* attempts to maximize the minimum normalized cell loss faced by any source.

A mixed–integer programming problem can be formulated to find solutions that satisfy either the *maxmin criterion* or the *minmax criterion.* Even though this is a mixed–integer programming problem, we will prove that a greedy sequential algorithm solves this problem. Strikingly similar proofs can be constructed for both the criteria. The existence of the greedy algorithm makes implementation of this scheme very simple.

Also, we conjecture that the normalized packet loss for each of the sources converges to a constant in the long–run. If this constant is larger than 1.0 for any source, then the QoS requirement for that source is not being satisfied. Otherwise, the packet loss faced by the source is less than its target and hence its QoS requirement is said to be satisfied.

Under some conditions, we conjecture that this constant is the same for all the sources in the system. If this conjecture is true and these conditions are met, then a value of at most 1.0 for this constant implies that with the available bandwidth, the QoS requirements of all the sources can be met. Also, the inevitable packet loss is distributed among the sources in a fair manner. If the value of this constant is greater than 1.0, then none of the QoS requirements are met. But the sources face the same degradation in the quality of their connection. An expression is derived for computing the value of this constant. This can be used to determine, *a priori*, whether accepting a new source into the system would violate the QoS requirements of the other sources present in the system.

Chapter 2

Multiaccess Communication

This chapter presents a review of multiple access protocols for packet–based communication systems. This review is neither complete nor exhaustive of all existing schemes. The reader is referred to Tobagi's excellent article [58] on multiaccess protocols for a more thorough and complete presentation.

The need for multiaccess protocols arises whenever a communication channel is shared by a number of independent users. The main issue is to control access to a common channel so as to efficiently allocate the available communication bandwidth to the many users. The solutions to this problem form the set of protocols known as multiaccess protocols. A multiaccess (or multiple access) protocol is a set of rules by which a number of distributed remote stations communicate over a shared channel.

The performance of multiaccess protocols is evaluated according to various criteria, most important of which are, high bandwidth utilization and low message delays (for packet data applications). It is highly desirable to have a multiaccess scheme with the ability to simultaneously support traffic of different types with a wide range of performance objectives. Robustness of the multiaccess protocol, defined as the insensitivity to errors resulting in misinformation [58], is also important.

There are two general approaches to transmitting information from one node to another within a subnetwork.

- In circuit switching, usually associated with the voice telephone network, each session is allocated a given transmission rate r. A path is then created from the source to the destination where each transmission link allocates a transmission rate r for this session. Any new session that is created does not have access to the above link. Thus, the entire link is dedicated to this session.

- In store–and–forward switching, usually associated with the Internet, each session is initiated without actually reserving any portion of bandwidth on any link. Bandwidth is provided on a demand basis rather than a fixed allocation. Usually, the sessions are never guaranteed any minimum rate or maximum transmission delay of their packets. These parameters depend on the current traffic in the network. The delays could depend on the number of nodes in the link and the congestion in each of these nodes.

Circuit switching is usually used for time–sensitive services which need a minimum (or fixed) transmission rate (voice networks). Circuit switching is rarely used for data networks since the data sources are bursty, *i.e.*, the data sources need access to the channel rather infrequently, and do not need a constant transmission rate all the time. In fact, for data sources, the peak–to–average ratio in the required data rate may be quite large. In order to satisfy this need with circuit–switching techniques, one must allocate the peak rate to a bursty source. This results in a very low channel utilization since the channel is not used during idle periods, when the data user has no information to transmit. Thus, circuit–switched techniques are not well suited for data applications. Store–and–forward switching is the choice for data networks.

2.1 Classification of multiaccess protocols

Tobagi [58] classifies all multiaccess techniques into five categories as follows :

- Fixed–assignment techniques consist of schemes that allocate a fraction of the available bandwidth to each user in a static fashion, independently of their activity. Circuit-switching techniques usually fall into this category.

- Random access techniques provide the entire channel to the user to be accessed in a random fashion. An active user, one with information ready to be transmitted, accesses the channel. But when the transmissions of more than one user occur simultaneously, a collision is said to occur (unless CDMA is used) and all or part of the information needs to be re–transmitted. Thus, collisions cause inefficient channel utilization. The problem then becomes one of controlling the access in such a way that high utilization can be achieved.

- Centrally controlled demand assignment techniques consist of a central authority that controls the bandwidth allocation based upon the users' demands for bandwidth. These techniques assume that explicit control information regarding the bandwidth requirements of the users is periodically transmitted to the central authority.

- Distributed demand assignment techniques differ from the centrally controlled techniques in that there is no central authority distributing bandwidth based on demands. These techniques also require that control information on the bandwidth requirements of the users be exchanged among them. Based on this information, all the users individually execute a distributed algorithm for channel access.

- Hybrid techniques consist of a mixture of one or more of the techniques from the above categories.

The multiaccess techniques can be further divided into two categories – "slotted" or "unslotted" – depending on whether time slotting is necessary for the specified operation. In the slotted versions of the protocols, time is divided into time slots and each transmission may begin only at the start of a time slot.

2.2 Fixed assignment techniques

Fixed assignment techniques allocate a portion of the bandwidth to each user in a static fashion without regard to the user's activity. These techniques are either *orthogonal*, like FDMA and TDMA, or *quasi–orthogonal* like CDMA.

- Frequency Division Multiple Access (FDMA) consists of assigning to each user a portion of the bandwidth and confining the user's access to this sub–band. Orthogonality is achieved in the frequency domain. It is a very simple protocol to implement and once the bandwidth assignments are made, it requires no coordination between the users. The total bandwidth, B, is divided into m channels, each with a bandwidth of B/m. The channel is usually the unit of allocation and bandwidth requirements of more than B/m can be satisfied by allocating more than one channel.

- Time Division Multiple Access (TDMA) allocates fixed pre–determined time slots to each user. The user gets access to the entire channel bandwidth during these time slots. Orthogonality is achieved in the time domain. In spite of the fact that TDMA is more complex to implement than FDMA, TDMA is preferred to FDMA in digital cellular systems like GSM, D–AMPS, etc. The main advantage of TDMA over FDMA is the fact that all receivers listen to the same channel while senders transmit over the same channel at different times. This facilitates better control and flexibility for bandwidth allocation.

- Code Division Multiple Access (CDMA), unlike FDMA and TDMA, lets all the users access to the entire bandwidth all the time. It achieves effective orthogonality by the use of different spreading codes, one for each user. The receiver has knowledge of the transmitter's code and uses this to decode the information. The code assigned to a user here is analogous to a channel in the above two techniques.

2.3 Random Access Techniques

Random access techniques deal with many distributed users who generate information that is bursty in nature. A sufficiently long idle period exists between two bursts of information to prevent allocating dedicated bandwidth to these users. So the users share a common high–bandwidth link and attempt to use the entire bandwidth once they have information to transmit. A collision between the transmissions of two or more users destroys all or part of the data. Hence, either the whole message or part of it needs to be re–transmitted which reduces the channel utilization. These techniques attempt to control access to the common channel in a fashion to achieve acceptable level of performance. In this section, we will briefly discuss some of the techniques that fall under this category.

2.3.1 ALOHA and its variants

ALOHA was first introduced by Abramson [1] to solve a multiaccess problem in a radio network at the University of Hawaii. Each node, upon receiving a packet, transmits it immediately. If the packet is involved in a collision, then the node attempts retransmission of that packet after a random delay. The user who transmits its packet receives an acknowledgment from the central node if there was no collision. If not, the user assumes that the packet was lost and backs off for a random amount of time before attempting retransmission. The random delay is introduced to avoid continually repeated collisions.

The main advantage of ALOHA is its simplicity. All the nodes base their actions only on the status of the feedback they receive on their recent transmission. It requires no interaction among the nodes and thus is very effective when the number of nodes is unknown or when no means exist to coordinate their actions. A major disadvantage is that it suffers from a stability problem. A multiaccess protocol is said to be stable for a given arrival rate if the expected delay per packet is finite [4]. ALOHA is unstable, in this sense, for any arrival rate greater than zero. Also, its maximum achievable throughput (with infinite mean delay) is $1/2e$. Although its maximum achievable utilization is low, ALOHA and its variants are superior to fixed assignment techniques when the users are very bursty in nature.

Slotted ALOHA

Slotted ALOHA is a variant of ALOHA in which time is divided into slots (the duration of a slot is equal to the packet transmission time), and all users are required to synchronize the start of their packet transmissions to coincide with the slot boundary. When two or more packets collide, they overlap completely rather than partially and this provides an increase in the maximum achievable utilization, which in slotted ALOHA is $1/e$, but, as in ALOHA, slotted ALOHA is unstable for any arrival rate greater that zero.

Stabilized slotted ALOHA

Attempts to stabilize slotted ALOHA were based on estimating the number of backlogged nodes, n [55, 22]. So when the estimate of n is higher, the attempt rate is lowered and vice–versa. These algorithms have shown that stability can be achieved when the arrival rate of packets is less than $1/e$.

Splitting Algorithms

This class of algorithms rely on different contention resolution techniques to improve upon the maximum stable throughput of slotted ALOHA. They are based upon the general idea of splitting the colliding nodes into subsets, one of which transmits in the next slot. If the collisions are not resolved, then further splitting takes place till all collisions are successfully resolved.

Tree algorithms [7] split the colliding nodes into two subsets (by flipping a coin, for example). The first subset transmits in the next slot, and if the slot is either idle or success, then the second subset transmits in the subsequent slot. In the event of a collision, the first of the subsets splits again, while the second waits for the resolution of that collision. A maximum throughput of 0.43 packets per slot was achievable with this algorithm. Another class of algorithms, called First–Come First–Served (FCFS) splitting [26, 37], was proposed and analyzed with achievable throughput of 0.4878.

2.3.2 Carrier Sense Multiple Access

One way to avoid collisions or at least to minimize the duration of the overlap of transmissions from two or more users is to listen to the channel before beginning the transmission and proceed with the transmission only if the channel is perceived to be in an idle state. This feature gives rise to a random access technique known as Carrier Sense Multiple Access (CSMA) [30]. Before beginning transmission, a user listens to the channel to ensure it is idle. Also, in the event of a collision (due to the transmission delays, the probability of a collision is non–zero), the users involved in the collision immediately halt their respective transmissions immediately and thus the duration of a collision is minimized. The halted transmissions are rescheduled according to the random delay.

There are two main CSMA protocols known as non–persistent and p–persistent CSMA depending on whether the transmission by a user which finds the channel busy is to occur later or immediately following the current one with probability p. In non–persistent CSMA, a user upon sensing an idle channel transmits immediately. But if the channel is sensed to be busy, the transmission is rescheduled according to a random delay. In p–persistent CSMA, the user transmits immediately if the channel is idle. If the channel is busy, then the user waits until the channel goes idle and then transmits immediately with probability p. With probability $1 - p$, the transmission is delayed by a time τ, called the minislot time, and if at this time the channel is perceived to be idle, then the above process is repeated. If the channel after a duration of τ is sensed to be busy, then the user reschedules its transmission according to a random delay (acts as though there was a collision). Ethernet is an example of a local communication network where CSMA has been implemented successfully. The 1–persistent CSMA is a special case of the p–persistent CSMA where $p = 1$. The maximum achievable throughput for CSMA schemes could be as high as 85%, but these schemes also suffer from instability.

In a radio environment, there is a possibility that two devices could be within range of each other's transmission, but due to factors like fading, they may not be able to sense each other's carrier. This decreases the efficiency of CSMA since the collisions could be longer than the maximum transmission delay. Many variants of CSMA have been proposed to work around this problem. Also, it is important to note that the stabilization algorithms discussed above could be used in CSMA and its variants also.

2.4 Centrally controlled Demand Assignment

The demand assignment techniques (both centrally controlled and distributed) require the exchange of information concerning the bandwidth requirements of all the users. In the centrally controlled schemes, a central authority receives the requests for bandwidth requirements and allocates bandwidth to the requesting users based on some service discipline.

2.4.1 Polling systems

A central controller sends polling messages to each user, one–by–one, asking them if they have information to transmit. If the user does have information to transmit, it does so immediately. If not, it sends a negative reply to the central station, which then sends the polling message to the next user on its polling list. Polling requires this constant exchange of control messages between the users and the central station and is efficient [31] only if

- the round–trip propagation delay[1] is negligible.
- the overhead due to the polling messages is low. Otherwise, the channel efficiency is lowered due to this overhead.
- the users are not bursty in nature.

Hayes [24] proposed a modification to the above scheme to reduce the high overhead involved in determining which users need to transmit. This modified polling scheme, known as probing, assumes that the central controller can broadcast to all terminals asking if any of them needs to transmit. The users could respond with a positive acknowledgment. In order to determine which users responded, the central station breaks the population down in to subsets (according to a tree structure) and repeats the question to the subsets. The process is continued until all the users having messages are identified. This is similar to the splitting algorithms described in section 2.3.1.

2.4.2 Split-Channel Reservation Multiple Access (SRMA)

In SRMA [59], the available bandwidth is divided into two channels – one is used to transmit control information while the second one is used for data. The users send the control messages containing their bandwidth requirements over the control channel

[1] This is the time interval between a user sending a message to another one and getting a reply from the latter.

using a random access technique (ALOHA or CSMA). The central controller queues the successfully received requests and sends a message on the control channel to the users informing them of the successful reception of their requests and the time when they can begin their transmission. Requests may be served according to FCFS or any other queuing discipline.

Resource Auction Multiple Access (RAMA) [3] is similar to SRMA, but the requests are not transmitted using ALOHA or CSMA. Instead, the control channel guarantees that there is at least one successful request getting across to the central controller in every time unit (assuming there exist users with messages to send).

2.5 Demand Assignment with distributed control

The basic element underlying all the distributed schemes is the need to exchange control information among the users, either explicitly or implicitly. Using this information, all the users independently execute the same algorithm resulting in some coordination in their actions. It is very essential that all users receive the same information regarding the requirements of the other users and the usage of the channel.

2.5.1 Reservation ALOHA

Reservation ALOHA [32] is an extension of slotted ALOHA. All users contend for the use of the channel in a fashion similar to slotted ALOHA. When a user succeeds in transmitting its packet in a particular slot, then this user is guaranteed the usage of the channel in subsequent slots for as long as the user needs to use it. When an idle slot is detected, then all users are eligible to contend, as in slotted ALOHA, for the use of the channel. The channel utilization in Reservation ALOHA is better than that in slotted ALOHA since one successful contention may result in more than one packet to be transmitted.

2.5.2 FIFO Reservation Scheme

Every so often, a time slot is divided into many minislots which are used to make reservations and also to transmit acknowledgments [58]. Each user is expected to maintain a count of the number of future slots that are reserved and the position of the slot at which it is to begin its own transmission. These positions are determined by a FIFO discipline based on successful reservations received. The duration of a reservation could be limited to prevent a single user from grabbing the channel for a long period of time resulting in large delays for the other users. Some additional information is included in the reservation minipacket in order to facilitate the correctness of the information maintained by all the users and also to let new users joining the system to acquire information about the current status of the system.

2.6 Hybrid schemes

One or more of the schemes from the above categories can be combined to yield a multiaccess scheme. Packet Reservation Multiple Access (PRMA) [19] is an example of a hybrid scheme. It is a combination of slotted ALOHA and TDMA. A detailed description of PRMA is presented in Chapter 3.

2.6.1 Priority–Oriented Demand Assignment

Another example of a hybrid scheme is Priority–Oriented Demand Assignment (PODA) [27]. It is an attempt to incorporate all the properties and advantages seen in many of the previous schemes. The channel time is divided into two subframes – an information subframe and a control subframe. The information subframe contains scheduled packets, which also contain piggybacked control information such as reservations and acknowledgments. The control subframe is used exclusively to send reservations.

Access to the control subframe can take many forms like

- TDMA if the number of stations is small, giving rise to FPODA (fixed–assignment PODA),

- contention based (like ALOHA) if the stations have a low activity factor, giving rise to CPODA (contention PODA), or
- a combination of the two.

Chapter 3

Frame Reservation Multiple Access

In order to support a high density of users, with limited radio spectrum, frequency reuse with low spatial separation (microcells) is essential. A dense grid of microcells will, inevitably, lead to an increased volume of network control traffic to provide fast mobile location and handoffs[1]. Goodman *et al.*[17] proposed a packet-switched architecture with interface units that route packets autonomously, thereby freeing network resources to handle the increased network control functions. The uplink multiaccess protocol of choice in [17] was Packet Reservation Multiple Access (PRMA). A comparison of multiple access schemes for packet radio networks is presented in [12] and PRMA comes out as an attractive candidate for mixed services over a range of cell sizes.

In this chapter, we propose modifications to PRMA, with a view to reduce receiver activity [49]. The resultant protocol is called Frame Reservation Multiple Access (FRMA). Also, PRMA has been a thorn in the flesh as far as exact analytical models to analyze its performance are concerned. Previous studies [39, 63, 48] have resorted to approximate analytical models. We construct an exact analytical model for FRMA to compute the performance measures of interest. We have proposed and compared different schemes for choosing the design variables. An optimization problem has been formulated and solved for one of the above schemes to choose design variables in order to optimize the performance of the system.

The rest of the chapter is divided as follows. An introduction to PRMA and the various models, found in the literature, to analyze its performance are presented in Section 3.1. A description of Frame Reservation Multiple Access (FRMA) can be found

[1]When a mobile terminal with an active connection moves from one cell (the area served by one base station) to another, the process of seamlessly transferring the connection from one base station to the other is referred to as a handoff or a handover.

in Section 3.2. Section 3.3 contains an exact analytical method to compute performance measures (voice packet dropping probability). In Section 3.4, we provide answers to the question *What is a good choice of the permission probability in every frame?* We present numerical results of the performance measures in Section 3.5.

3.1 Packet Reservation Multiple Access

Conversational speech consists of alternating periods of talk and silence. In conventional circuit–switched telephone networks (and also in TDMA), at the start of a voice call, a circuit (or channel) is allocated for the entire duration of the call. The channel bandwidth is not utilized by the voice source during periods of silence in the conversation.

Goodman *et al.*[19] proposed Packet Reservation Multiple Access (PRMA), an uplink multiaccess protocol for packet voice terminals in microcellular systems. PRMA exploits the silence periods in conversational speech to achieve statistical multiplexing of more voice terminals than the number of available channels, thereby providing a capacity gain over TDMA. Each voice terminal is assumed to be equipped with a voice activity detector [6], which separates speech into talkspurts and silence periods. Thus during silence periods, a voice terminal gives up its channel for possible use by other voice terminals. PRMA has *soft* capacity, *i.e.* voice quality degrades gracefully as the number of voice terminals is increased beyond the designed number. In contrast, TDMA has *hard* capacity, *i.e.* the number of conversations is limited to the number of channels available.

The channel in PRMA is slotted. Slots are grouped together into frames, with the frame rate being identical to the rate of generation of speech packets. The slot period is equal to the transmission time of a speech packet. That is, an active voice source generates one packet per frame. To describe the system we adopt the following definitions.

- R_c, the channel rate, equals the number of bits that can be transmitted per unit time,

- R_s, the source rate, is the number of bits that a speech source generates while in a talkspurt,

- O_h, the packet overhead, is the number of the bits in the header of each packet that includes, among others, the identity of the destination of the packet so it can be routed appropriately. It might also have some form of error protection for the header.

- T represents the frame duration measured in seconds.

From these definitions, the number of slots per frame, N, is given by

$$N = \frac{R_c T}{R_s T + O_h}. \tag{3.1}$$

We assume throughout that N is an integer.

The protocol is summarized as follows:

- Every slot is either reserved by a voice terminal or available for contention by the contending terminals.

- If the terminal that holds the reservation for a slot has a packet to transmit, it does so in its reserved slot. It signals the end of its talkspurt by not transmitting any packet in its reserved slot. This slot is then classified as available for contention in subsequent frames.

- Terminals with packets to transmit but without a reservation are called contending terminals. They contend for available slots by transmitting their oldest packet in any available slot with probability p called the permission probability. The base station broadcasts a feedback after every slot informing the terminals of whether it was able to decode correctly a packet in that slot. If the base is able to decode the header of the packet in a slot, it identifies the terminal that transmitted it and this terminal obtains a reservation for that slot in subsequent frames. Otherwise, a NULL acknowledgment is broadcasted, in which case that slot is available for contention. A NULL acknowledgment could be a result of either no transmission or a collision in that time slot (or a result of transmission errors that prevent a collision–free packet to be decoded correctly).

PRMA can be viewed as a combination of slotted ALOHA and TDMA. As mentioned above, the speech terminal is assigned a channel (in the form of a slot in the PRMA frame) only during talkspurts and gives up the use of that channel during silence periods. Contending terminals use slotted ALOHA in the available slots to request a reservation. That is, all contending voice terminals transmit a packet in an available slot with probability p called the permission probability. Once the contention is successful, the speech terminal is assigned that time slot in subsequent frames for its exclusive use (a reservation) for the duration of the rest of its talkspurt. Thus, PRMA behaves like TDMA when a terminal has a reservation.

Speech packets have a finite lifetime and lose their significance if they are delayed beyond a certain time. Since packets require prompt delivery, voice terminals are equipped with a finite FIFO buffer to store speech packets. Packets arrive once every frame and if the buffer is full, the oldest speech packet is dropped to make room for the newly arrived speech packet. The number of speech packets in a talkspurt that are dropped depends on the time required, from the start of the talkspurt, for the speech terminal to contend successfully and obtain a reservation. Congestion in the contending state leads to access delays and hence higher voice packet dropping and degradation of speech quality. The system performance measure of interest is, therefore, voice packet dropping probability. A voice packet dropping probability of at most 1% is considered acceptable for good speech quality [39]. $M_{0.01}$, defined as the capacity of a PRMA system, is the maximum number of voice terminals that can be supported with a voice packet dropping probability of at most 1%.

Nanda *et al.*[39] analyze the performance of a voice–only PRMA using Equilibrium Point Analysis (EPA). They model the system as a Markov Chain, but note that even though the number of state variables is $N+3$, the number of the states in the chain is $O(M^2 2^N)$ where M is the number of voice terminals and N is the number of slots per frame. EPA [57] is an analytical technique that approximates the probability distribution of the value of each state variable by an impulse of weight 1 at one point, called the equilibrium value of that variable. Equating the inflow and the outflow at each state gives a set of equations solving which the equilibrium value of each state variable can be obtained. Even though the correctness of EPA has not been proved, it seems to be a useful (and popular) technique to analyze systems where the state space is too large for exact computation. Using these equilibrium values, analytical expressions that approximate various performance measures like voice packet dropping probability and system throughput have been derived.

A phenomenon similar to the instability of slotted ALOHA has been observed here where there are periods when the number of contending terminals grows to a large number resulting in a lot of collisions and higher packet loss. Good choices for the permission probability have been suggested to avoid undesirable equilibrium points.

There are times when all the slots in the frame are reserved. During these times, no contention is possible and all the contending terminals are forced to drop packets. This could also lead to undesirable equilibrium points where the number of contending terminals grows and when slots become available, a lot of collisions and hence higher packet loss result. Consequently, Mitrou *et al.*[36] proposed a variant of PRMA with two notable modifications.

- In PRMA, any slot could carry information or control packets depending on whether they are reserved or available respectively. A terminal that contends successfully in an available slot gets a reservation for that slot and hence that slot becomes an information slot for the duration of that talkspurt. Mitrou *et al.*propose to set aside a few slots meant to be used as purely contention slots. Thus, the number of reserved slots is never allowed to reach N. Those terminals that contend successfully when there are no information slots available are queued and assigned a reservation as soon as an information slot becomes free. This reduces the load on the contention queue and avoids undesirable equilibrium points [39].

- In PRMA, a contending terminal attempts to make a reservation by transmitting its oldest packet in an available slot. In the event of a collision, the whole slot is wasted. Mitrou *et al.*[36] proposed to divide any contention slot into minislots so that contending terminals transmit just a control packet instead of an actual information packet to contend. This increases the access delay for the first packet, but improves the performance of the contention process.

Wu *et al.*[63] propose a Markov model to analyze PRMA with the assumption that state transitions occur at frame boundaries. They compute the transition probability matrix of an imbedded Markov Chain using an iterative procedure and claim that the results of their model are more accurate than those in [39].

Qi and Wyrwas [48] present another methodology for the analysis of PRMA using a mixture of a Markov Chain and equilibrium point analysis. They use EPA to compute equilibrium values for the state variables. They use these equilibrium values of the state variables in computing the steady–state probabilities of a Markov Chain and then go on to compute the performance measures of interest (voice packet dropping probability, system throughput). They have also proposed adaptive permission probability schemes, which are based on estimates of the number of contending terminals.

Bianchi *et al.*[5] describe another variant of PRMA, called Centralized–PRMA (C–PRMA), where the base station uses information on time–of–expiration of the packets (this is the time when the packet will be dropped from the source buffer) to schedule transmissions from mobile terminals. The base station also schedules control slots in the frame and, as in [36], minislots are used for contention. Information on packet expiration times are transmitted in the contention packet to enable scheduling at the base station. This protocol improves throughput and performance under fading conditions, but requires more intelligence and computing power at the base station.

Goodman and Wei [20] study, using simulations, the various factors that affect the performance of PRMA. Their results for a 20–slot system, with identical system parameters as those of [39], agree well with the analytical results of [39]. This study concludes that for this 20–slot system, the maximum number of simultaneous conversations, while maintaining the voice packet loss under acceptable limits, is 37 (corresponding to 1.65 conversations per channel). They also conclude that for a wide range of packet sizes (8 ms per packet to 34 ms), and for all systems with 16 or more channels (i.e., channel rate of at least 512 kbps with a source rate of 32 kbps), the number of conversations per channel was 1.5 or more. Other factors studied in this paper are the sensitivity of the speech activity detector, the permission probability of the contention scheme, and the maximum time delay for the transmission of the speech packets.

All of the above studies assumed that the number of voice conversations in the system is constant at any given time. Xie and Goodman [64] present an analytical study of PRMA with a variable number of voice calls. They use an M/M/c model where c is the capacity of a PRMA system with a particular set of system parameters. When the number of calls in the system is c, then any newly arriving calls are blocked.

3.2 Frame Reservation Multiple Access

3.2.1 Motivation

In PRMA, the base station broadcasts, to all the mobile terminals, a short feedback packet based on the information it received in the previous slot. If the base station is able to identify the header of the packet it "received", it identifies the terminal that transmitted it and the slot is reserved for that terminal in future frames. If the base station is unable to decode the header of the packet, then it transmits a "null" feedback and the slot is considered available for contention in future frames. Thus, all the mobile terminals are expected to listen to this feedback message in every slot to obtain the status of each slot in the frame. Hence, the receiver at the mobile terminal is always active.

Practical considerations may prevent the base station from being able to transmit feedback (acknowledgment) packets immediately at the end of the slot. In fact, in one practical radio, Ziegler and Pollini [67] reported that there may be a delay of about half a frame before the base station can broadcast its acknowledgment message. Narasimhan and Goodman [40] extended the EPA analysis of PRMA in [39] to account for the delay in the acknowledgment packets and its effect on system performance. In their model, the terminals that contend for a slot wait for the acknowledgment to learn the result of their contention. If the contention was unsuccessful they return to the contention state. Thus the frequency of contention is reduced due to the wait for acknowledgment. So the voice packet dropping probability increases marginally due to the reduced rate of contention. But at the same time, the terminals are also more *polite* in not attempting to contend for every available slot. This reduces the occurrence of undesirable equilibrium points and increases the stability of the system. Also, some of the effect of the acknowledgment delay can be compensated for by an increase in the permission probability.

Rauber [49] studied the issues in designing receivers for mobile terminals in a PRMA system and concluded that the receiver activity should be reduced in order to extend battery life. Moreover, Rauber also suggested that there are advantages to time–division duplex (TDD) uplink and downlink transmissions. In an attempt to incorporate the results of the study in [49], we propose modifications to PRMA. The resulting protocol is called Frame Reservation Multiple Access (FRMA).

In FRMA, the base station broadcasts an acknowledgment packet at the end of the uplink frame, rather than at the end of every slot as in PRMA. This reduces the need for the receiver at the mobile terminal to be switched on in every slot. Apart from extending battery life, this also permits mobile assisted handoffs by enabling the mobile terminal to listen to base stations in neighboring cells to identify potential candidates for a handoff (if necessary). The FRMA protocol is well suited for TDD operation of uplink and downlink transmissions (Figure 3.1). The base station is not expected to broadcast an immediate acknowledgment packet, thereby giving it sufficient time to process the information it received in the previous uplink frame.

3.2.2 Description of FRMA

The FRMA protocol is described below:

- The base station broadcasts between the end of one uplink frame and the beginning of the next uplink frame (Figure 3.1), the identity of the slots which are reserved and the identity of the terminals that hold these reservations. All other time slots are considered available.

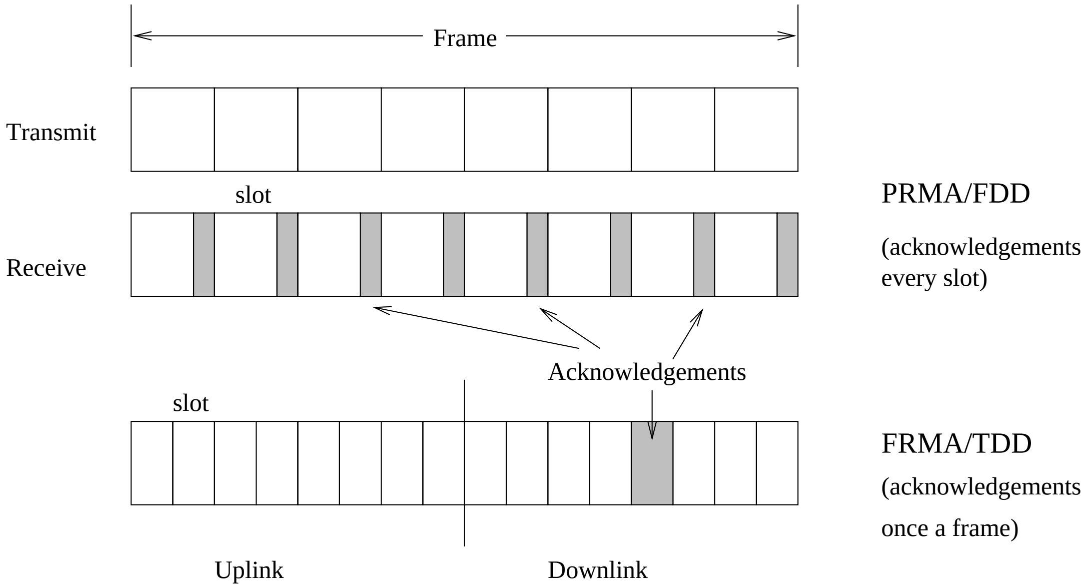


Figure 3.1: PRMA and FRMA frame structures

- If a terminal has a reservation for a time slot in the frame, it transmits a packet in that time slot. Otherwise, the terminal can contend for each of the available time slots with probability p, termed the permission probability.

- In any time slot, if exactly one terminal transmits a packet, this transmission is successful and the terminal could obtain a reservation for this time slot. If two or more terminals transmit in the same time slot (or if none of the terminals transmits), then these transmissions (if any) are unsuccessful and the packets are assumed lost. In the event this happens, that time slot will be classified as available in the next frame.

- Unlike PRMA, the terminals have to wait until the end of the frame before being informed of the result of their transmissions.

- It is possible that the same terminal successfully contends for more than one time slot in the frame, in which case, the base assigns one of these time slots to this terminal in an arbitrary manner. Thus, at the end of a frame, each terminal can hold a reservation for at most one time slot.

- If a contending terminal fails to obtain a reservation in a frame, then it drops the oldest packet from its buffer. Therefore, the speech packets that are not transmitted successfully in a frame are dropped at the end of the frame.
- The permission probability, p, stays constant throughout a frame. But the value of p is not necessarily constant across frames as it can be varied from one frame to the next.
- Terminals which are in the silent state at the beginning of a frame and enter a talkspurt in the middle of a frame wait till the beginning of the next frame before contending for a time slot.
- Under certain conditions, when more than one mobile terminal contends for an available slot, it is possible for the base station to correctly decode a packet from one of the contending mobile terminals. This phenomenon is called capture. In our model, we assume that no capture phenomenon occurs in the system. This results in a more conservative estimate of the system performance since ignoring the effect of capture results in more collisions than otherwise.
- The channel is assumed to be error–free and the transmission delays are assumed to be negligible.

3.3 Analysis of FRMA

In this section, we construct a Markov Chain model of a voice–only FRMA system, assuming the number of voice terminals to be a constant. We derive the transition probability matrix of the Markov Chain and compute the performance measure of voice packet dropping probability.

3.3.1 Speech Model

A speech source creates a pattern of talkspurts and silent gaps, as classified by a speech activity detector [6]. The speech activity detector is modeled as a two–state Markov process (Figure 3.2). We assume that the frame duration is so short that the probability

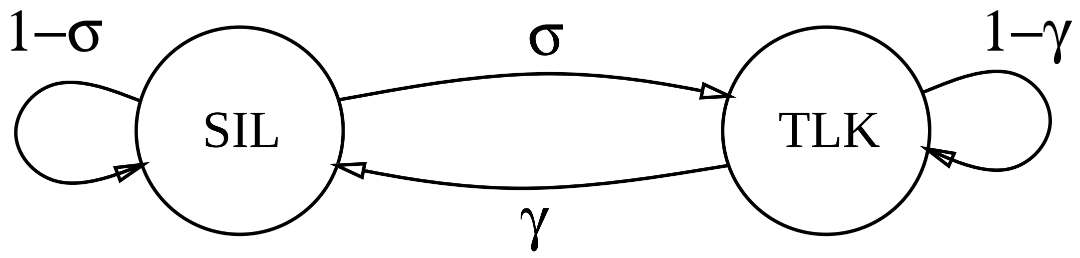


Figure 3.2: Two–state Markov model of the speech activity detector

of more than one transition per terminal (from TLK to SIL or vice–versa) is negligible. The probability that a talkspurt with mean duration t_1 ends in a frame of duration T is

$$\gamma = 1 - e^{-T/t_1}. \tag{3.2}$$

This is the probability that a terminal in a talkspurt (state TLK) makes a transition to the silent state (SIL) in any given frame. Correspondingly, the probability that a silent gap, of mean duration t_2, ends during a frame is

$$\sigma = 1 - e^{-T/t_2}. \tag{3.3}$$

We define the speech activity factor, α, to be the proportion of time a speech terminal is in the TLK state. Then,

$$\alpha = \frac{\sigma}{\gamma + \sigma}. \tag{3.4}$$

3.3.2 The Markov Chain

Let M denote the number of voice terminals in the system, and N denote the number of slots per frame (equation (3.1)). In this section, we show that the voice–only FRMA system can be modeled as a Markov Chain with its state space described by the vector (R, C), where

- R is the number of reserved slots at the beginning of a frame, and
- C is the number of contending voice terminals.

Any feasible state (R, C) must satisfy

$$\begin{aligned} 0 \leq \quad R \quad &\leq \min(M, N), \\ 0 \leq \quad C \quad &\leq M - R. \end{aligned} \tag{3.5}$$

Let S denote the set of feasible states (R, C).

The state transitions are assumed to occur between the end of one frame and the beginning of the next. Suppose at the beginning of the n-th frame, $R^{(n)}$ was the number of reserved slots, and $C^{(n)}$ was the number of contending terminals, then

$$\begin{aligned} R^{(n+1)} &= R^{(n)} + N_{CR}^{(n)} - N_{RI}^{(n)}, \\ C^{(n+1)} &= C^{(n)} - N_{CR}^{(n)} - N_{CI}^{(n)} + N_{IC}^{(n)}, \end{aligned} \tag{3.6}$$

where

$N_{CR}^{(n)}$ is the number of contending terminals that succeed in obtaining a reservation at the end of the n-th frame. The conditional distribution of $N_{CR}^{(n)}$ needs to be derived.

$N_{RI}^{(n)}$ is the number of terminals that had a reservation at the beginning of the n-th frame and finished their talkspurt during the n-th frame and hence moved to the silent state. This random variable, conditioned on $R^{(n)}$ and $C^{(n)}$, is distributed as Binomial$(R^{(n)}, \gamma)$.

$N_{CI}^{(n)}$ is the number of contending terminals that finished a talkspurt during the n-th frame and moved to the silent state. This random variable, conditioned on $R(n)$ and $C^{(n)}$, is distributed as Binomial$(C^{(n)} - N_{CR}^{(n)}, \gamma)$.

$N_{IC}^{(n)}$ is the number of silent terminals that began a talkspurt during the n-th frame and hence enter the contending state. This random variable, conditioned on $R(n)$ and $C^{(n)}$, is distributed as Binomial$(M - R^{(n)} - C^{(n)}, \sigma)$.

The number of states is finite, and for any given value of the permission probability, $0 \leq p \leq 1$, the state $(0, 0)$ is reachable from every other state. Thus, we have a single communicating class, and the system can be modeled as a Markov Chain with steady–state probabilities $\pi(R, C)$. The number of states in this Markov Chain is given by

$$|S| = \sum_{r=0}^{\min(N,M)} M - r + 1 = \begin{cases} \frac{(M+1)(M+2)}{2} & M \leq N \\ (N+1)(M+1-\frac{N}{2}) & M > N. \end{cases} \tag{3.7}$$

In order to derive the conditional distribution of $N_{CR}^{(n)}$, consider an arbitrary frame, at the start of which the system state is described by (R, C), *i.e.* there are R reserved slots (or $N - R$ available slots) and C contending terminals. Let p be the permission probability to be used by the contending terminals. A successful contention occurs in any slot with probability, q, given by

$$q = Cp(1-p)^{C-1}, \tag{3.8}$$

and a failure (idle slot or collision) occurs with probability $1 - q$. Let J denote the number of available slots where there is a successful contention. Note that $N_{CR} \leq J$, since a contending terminal may have more than one successful contention. Since the outcome in one available slot is independent of the others, J has a binomial distribution given by

$$P\{J = j\} = \binom{N-R}{j} q^j (1-q)^{N-R-j}, \qquad j = 0, \ldots, N-R. \tag{3.9}$$

Since $N_{CR} \leq J \leq N - R$, the probability distribution of N_{CR} can be expressed as

$$P\{N_{CR} = s\} = \sum_{j=s}^{N-R} P\{N_{CR} = s | J = j\} P\{J = j\}. \tag{3.10}$$

We derive the expression for $P\{N_{CR} = s | J = j\}$ in Appendix A.

Speech packets that are not transmitted successfully in a frame are dropped at the end of the frame. The long–term packet dropping rate is computed as the ratio of the average number of packets dropped in a frame to the average number of packets generated in a frame.

Since contending terminals that fail to obtain a reservation at the end of a frame drop a packet each[2], the number of packets dropped at the end of the frame equals the

[2] Nanda *et al.*[39] assumed that packets are dropped after a delay of $D_{max} = 40ms$, which was equal to duration of two frames. Our estimate of packet dropping probability is more conservative since in this case D_{max} is equal to the duration of one frame.

number of terminals that are blocked, *i.e.* fail to obtain a reservation. Each terminal that holds a reservation at the beginning of a frame successfully transmitted a packet in the previous frame since any terminal with a successful transmission in a frame continues to hold the reservation in the next frame. Hence, R is the number of successful packets in the previous frame and if D is the number of packets that are dropped, their sum $R + D$ must equal the number of packets generated during the previous frame. Hence, the packet dropping probability is given by

$$p_{drop} = \frac{E[D]}{E[R+D]}. \tag{3.11}$$

A contending terminal fails to make a reservation in an arbitrary unreserved slot with probability $1 - p(1-p)^{C-1}$. A contending terminal drops a packet at the end of a frame if it is not successfully transmitted in any of the $N - R$ available slots. Since the outcomes of the $N - R$ available slots are independent, a contending terminal drops a packet with probability

$$\delta(p, R, C) = (1 - p(1-p)^{C-1})^{N-R}. \tag{3.12}$$

Summing over all the C contending terminals, the expected number of dropped packets becomes

$$E[D|R, C] = C\delta(p, R, C). \tag{3.13}$$

Averaging over the stationary distribution of the system, the expected number of dropped packets per frame is

$$E[D] = \sum_{R=0}^{\min(M,N)} \sum_{C=0}^{M-R} E[D|R, C]\pi(R, C). \tag{3.14}$$

We observe that the number of terminals in the TLK state is binomially distributed and the expected number depends only on the number of terminals in the system and the speech activity factor, α (equation (3.4)). Hence, $E[R + D]$ is independent of the equilibrium probabilities of the Markov Chain. Thus, the packet dropping probability is directly proportional to the expected number of dropped packets per frame.

3.4 Choosing permission probability

Nanda *et al.*[39] assumed that the permission probability is held constant in every available slot. Also, they estimated that $p = 0.3$ is a *good* value for the permission probability for safe operation (avoiding undesirable equilibrium states). In this section, we propose the question *What is a good value for the permission probability?* and provide answers to it.

From section 3.3, we know that the state of the FRMA system is described by (R, C), where R is the number of reserved slots at the beginning of that frame and C is the number of contending terminals. At the start of every frame, we have complete knowledge of R, but C is unknown. Given that partial state information is available in every frame, can we use this information to improve the system performance compared to using a constant value for the permission probability?

3.4.1 Perfect Information

The number of contending terminals in any frame is an unknown quantity. So, in practice, we do not have perfect information on the system state. Suppose we assume that we do know with certainty the number of contending terminals in every frame. We choose permission probability in each frame so that the expected number of packets dropped in that frame is minimized. We conjecture that doing so does minimize the long–term packet dropping probability. In this regard, it is also worth noting the optimum permission probability in ALOHA has the same structure. If our belief is true, then this scheme gives us a lower bound on packet dropping probability for all FRMA systems operating under the same system parameters. From equation (3.12), the value of p that minimizes the expected number of dropped packets, in a frame that starts with (R, C) as the system state, is

$$p^*(R, C) = \begin{cases} 0, & C = 0 \\ \frac{1}{C}, & C = 1, 2, \ldots, M - R. \end{cases} \tag{3.15}$$

Notice that the optimal p^* is independent of the value of R. If $C = 0$, then there are no contenders and hence the value of permission probability is irrelevant.

While computing the transition probabilities of the Markov Chain, we use the permission probability given by Equation (3.15). It should be noted that achieving the performance of this scheme is almost impossible in practice. However, this scheme does provide us with a measure that could be used to evaluate the performance of a real system by comparing it with the performance of this ideal system.

3.4.2 Estimating the number of contending terminals

The number of contending terminals, C, is not known with certainty. Suppose we assume, at the beginning of frame n, we have an estimate of the probability distribution of C, $P_C^{(n)}(c)$. Given this distribution, we could choose a value of permission probability for the frame that minimizes the expected number of dropped packets. At the end of the frame, we observe the number of contending terminals that contended successfully and then update our estimate of the probability distribution of the number of contenders using Bayes' rule. Accounting for any new arrivals into the contending state and any departures from the contending state to the idle state, we could then obtain, $P_C^{(n+1)}(c)$, an estimate of the probability distribution of the number of contending voice terminals at the start of frame $n+1$. In this section, we describe the details of this algorithm.

At the start of the n-th frame, let $R^{(n)} = r$ represent the number of reserved slots. Let the probability distribution of the number of contending terminals, $C^{(n)}$ be

$$P_C^{(n)}(c) = P\Big\{C^{(n)} = c\Big\}, \qquad c = 0, 1, \ldots, M - r. \tag{3.16}$$

The value of permission probability, p, that minimizes the expected number of packets dropped can be obtained by solving

$$\begin{array}{ll} \min & \sum_{c=0}^{M-r} c(1-p(1-p)^{c-1})^{N-r} P_C^{(n)}(c) \\ \text{subject to} & 0 \leq p \leq 1. \end{array} \tag{3.17}$$

We use the convention that if $p = 1$ and $c = 1$, then $c(1 - p(1-p)^{c-1}) = 0$.

At the end of the n-th frame, we can observe the number of contending terminals that contend successfully. Using this information, we could update our estimate of the probability distribution of the number of contending terminals. For example, if we know that number of successful terminals is s, then there must have been at least s contending terminals at the start of the n-th frame. Let $S^{(n)}$ denote the number of contending terminals that succeed in getting a reservation. We know that $S^{(n)} \leq \min(C^{(n)}, N - R^{(n)})$. If $S^{(n)} = s$, then $C^{(n)} \geq s$. Thus we could *correct* our estimate of $P_C^{(n)}(c)$ by a Bayesian update. If $Q_C^{(n)}(c)$ denotes the *corrected* estimate of the probability distribution of the number of contending terminals at the start of the n-th frame, then

$$\begin{aligned} Q_C^{(n)}(c) &= P\Big\{C^{(n)} = c | S^{(n)} = s\Big\} \\ &= \frac{P\Big\{S^{(n)} = s | C^{(n)} = c\Big\} P_C^{(n)}(c)}{\sum_{i=0}^{M-r} P\{S^{(n)} = s | C^{(n)} = i\} P_C^{(n)}(i)}. \end{aligned} \tag{3.18}$$

In order to compute our estimate of the probability distribution of the number of contenders at the start of the $(n+1)$-th frame, we have to account for the terminals that leave the contending state after dropping the last packet of their current talkspurt and those that move from the idle state to the contending state due to the start of a new talkspurt. Thus,

$$P_C^{(n+1)}(c') = \sum_{c=s}^{M-r} \sum_{d=0}^{c-s} Q_C^{(n)}(c) B(C - s, \sigma, d) B(M - r - s, \gamma, c' - c + s + d), \tag{3.19}$$

where $B(n, p, x)$ is given by

$$B(n, p, x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x}, & x = 0, 1, \ldots, n, \\ 0, & \text{otherwise.} \end{cases} \tag{3.20}$$

$B(C - s, \sigma, d)$ is the probability that d out of $C - s$ terminals complete their talkspurt in the contention state and hence move to the SIL state. $B(M - r - s, \gamma, c' - c + s + d)$ is the probability that $c' - c + s + d$ of the $M - r - s$ terminals in the idle (SIL) state begin a new talkspurt and hence move to the contention state. Using $P_C^{(n+1)}(c)$ as the estimate of the probability distribution of the number of contenders at the start of the $(n+1)$-th frame, we perform the iteration for the next frame by solving the optimization problem in equation (3.17). Assuming that all the terminals are in the idle state at the start of frame 0, we could start with

$$P_C^{(0)}(c) = \begin{cases} 1, & c = 0, \\ 0, & c = 1, 2, \ldots, M. \end{cases} \tag{3.21}$$

This method is computationally intensive and hence is not well-suited for real–time implementation. Even though the optimization problem in Equation (3.17) is one–dimensional, the nature of the objective function makes it difficult to solve with a quick and easy procedure. The Bayesian update procedure is quite expensive computationally too because there is no closed form expression for $P\left\{S^{(n)} = s|C^{(n)} = c\right\}$. On the whole, this method is only of theoretical interest and is useful, again, for comparison purposes with other methods of choosing the permission probability.

3.4.3 Permission probability based on the number of reserved slots

As noted earlier, the number of contending terminals is usually not known with certainty. The computational cost of maintaining a good estimate of the number of contending terminals is quite expensive. Suppose we used only the information about the system state that is available to us, how does the system performance compare with that of the ideal system as described in Section 3.4.1. In this section, we describe another way of choosing the permission probability that is virtually free of computational effort. We formulate an optimization problem to minimize the packet dropping probability.

At the start of every frame, the number of reserved slots, R, is known with certainty. Given that this is the observable portion of the system state, we propose to choose permission probability as follows. Let $\mathbf{p} = (p_0, p_1, \ldots, p_{N-1}), 0 \leq p_i \leq 1, i = 0, 1, \ldots, N-1$ represent a vector of permission probabilities. The permission probability in any frame is chosen to be p_r, where r is number of reserved slots. This gives us a simple scheme of choosing the permission probability without heavy computation.

The permission probability vectors could either be stored at every mobile terminal or be broadcasted at the start of every frame by the base station. The disadvantage of the former method is that it is less flexible since any change in the vectors would require reprogramming the mobile terminals. But broadcasting the permission probability over the air interface in every frame would require sufficient protection against errors in the radio channel.

Given two permission probability vectors, $\mathbf{p}$ and $\mathbf{p}'$, the vector that results in a lower packet dropping probability is prefered to the other. Hence, the best choice of a permission probability vector is one that minimizes the voice packet dropping probability. The problem of determining the best vector can be formulated as

$$\begin{array}{lll} \min & \sum_{r=0}^{\min(N,M)} \sum_{c=0}^{M-r} c(1-p_r(1-p_r)^{c-1})^{N-r}\pi(r,c) & \\ \text{subject to} & 0 \leq p_r \leq 1, r = 0, \ldots, \min(N,M)-1. & (3.22) \end{array}$$

The objective function is the expected number of packets lost per frame. It is sufficient to minimize this quantity in order to minimize the packet dropping probability. The packet dropping probability, as previously defined, is the ratio of the expected number of packets dropped to the expected number of packets generated. But the latter quantity is independent of the contention process and, hence, the value of the permission probability. Hence, we could omit it from the objective function and minimize the expected number of packets dropped. This is a non–linear programming problem with the N–dimensional unit cube as the constraint set. This problem is difficult to solve using methods that require derivatives since the stationary probabilities, $\pi(r,c)$, depend on the particular choice of the permission probability vector.

We use a direct–search technique [25, 34] to solve the problem. The search can be initialized at any feasible point. At every iteration, we try to direct the search in a direction that decreases the value of the objective function. The current point is perturbed by a small amount in each coordinate (and the value of the objective function is computed for each perturbation) to determine a *good* direction in which to move the search. A new search point is obtained by moving in the selected direction in steps of a pre–determined size until this direction no longer decreases the value of the objective function. At every step, a check is enforced to ensure the feasibility of the new search point, which in this case is as simple as searching only in the N–dimensional unit cube. This process is repeated until the change in the value of the objective function between two iterations is less than a specified value (10^{-6} was the value used in our computation). This technique does not guarantee a global optimum. The solutions obtained could be local minima. But, as discussed below, we have reason to believe that the objective function is unimodular in the constraint set and, hence, has a unique minimum. The structure of the optimal solutions is discussed in section 3.5.

The main difference between this method and the one described in section 3.4.2 is that given a value of the number of reserved slots, we assume a probability distribution as the conditional distribution of the number of contenders computed from the steady–state probability distribution of the Markov Chain.

$$P\{C=c|R=r\} = \frac{\pi(r,c)}{\sum_{c=0}^{M-r}\pi(r,c)}, \qquad c=0,1,\ldots,M-r \tag{3.23}$$

Since this distribution is known in advance, we could compute the best value of permission probability *a priori*. Thus, the problem in equation (3.22) needs to be solved off–line once. But in the method described in section 3.4.2, since we estimate and update the probability distribution in every frame, we have to solve the optimization problem in every frame.

3.4.4 Constant permission probability

Most previous studies [39, 63, 28, 20] use a constant permission probability in all frames. This scheme ignores the available information about the system state at the end of each frame. The non–linear programming problem (equation (3.22)) can be reduced to a one–dimensional problem to choose the optimal value for the permission probability that can be used in the constant permission probability scheme.

3.5 Numerical results

We have evaluated the performance of three of the four schemes, to choose permission probability, mentioned above and present the numerical results in this section. The three schemes are

- Choosing permission probability under perfect information – this gives us a lower bound on voice packet dropping probability for all FRMA systems,
- choosing permission probability based on the number of reserved slots; we will refer to this scheme as the adaptive permission probability scheme, and
- the constant permission probability scheme

Table 3.1 summarizes the system parameters that we used to compute the performance metrics. The transmission rate of the radio channel was 432 Kbps. The speech sources generate information at 32 Kbps. With a frame duration of 11 ms, each active speech source generates 352 bits of speech information per frame. Each speech packet contains header information (80 bits long), which includes, among other things, the address of the destination of the packet. This results in 432 bits per packet. In a frame of 11 ms, a channel rate of 432 Kbps can accommodate 4752 bits which amounts to 11 packets (and hence 11 time slots per frame).

Source rate	32 Kbps
Channel rate	432 Kbps
Overhead per packet	80 bits
Frame duration	11 ms
Number of slots per frame	11
Average talkspurt duration	1 sec
Average silence duration	1.35 sec

Table 3.1: FRMA system parameters

Figure 3.3 shows the voice packet dropping probability for the three schemes under consideration. As defined before, the capacity of the system, $M_{0.01}$, will be the maximum number of simultaneous voice conversations that can be supported while maintaining the voice packet dropping probability below 0.01. The capacity of a PRMA system operating under the parameters given in Table 3.1, with a constant permission probability of 0.3, is 17 simultaneous speech conversations. The capacity for the FRMA system using a fixed permission probability is 17 users. This implies that, under the given operating parameters, there is no loss in capacity due to FRMA, *i.e.* the capacity of the FRMA system with a fixed permission probability scheme is equal to that of an equivalent PRMA system. Using the adaptive scheme, where we choose the permission probability for a frame based on the number of reserved slots at the start of the frame, we can support 18 simultaneous voice conversations while maintaining the voice packet dropping probability within acceptable limits. Thus, we can improve on the capacity of the system by using the adaptive permission probability scheme instead of a fixed

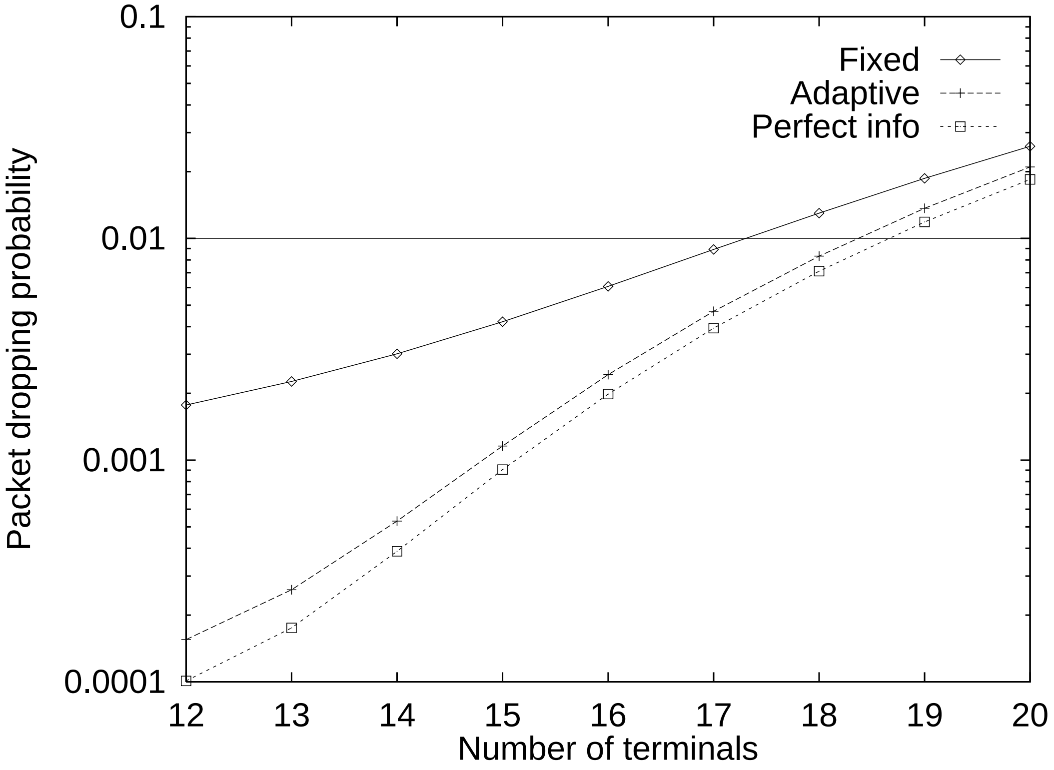


Figure 3.3: Comparison of the packet dropping probability under the three schemes under consideration.

permission probability scheme. Also, the ideal scheme tells us that we can support a maximum of 18 conversations under the given system parameters.

It is interesting to note that the packet dropping probabilities achieved using the adaptive scheme are quite close to the lower bound. The optimal solutions for the adaptive scheme were computed using a direct–search technique. Many instances of the search were initiated from numerous random points in the N–dimensional unit cube. The search was terminated when the incremental gain in the value of the objective function was below a certain tolerance factor specified as an input parameter. The solutions obtained by each one of these instances were found to be very close to each other. This leads us to conjecture that we converge to a global optimum in each one of the instances and that the objective function is unimodal in the N–dimensional unit cube.

Also, note that the packet dropping probabilities approach each other as the number of voice conversations is increased. This implies that the performance of the constant permission probability scheme approaches that of the adaptive permission probability scheme as M, the number of voice terminals in the system, increases. This is reflected in the structure of the optimal solutions, shown in Figure 3.4. As M increases, the optimal solution for the adaptive scheme does not change much with R, the number of reserved slots. Thus, its performance should be closer to that of the fixed scheme. On the other hand, when M is small, using the additional information of the number of reserved slots to choose the permission probability should achieve better performance than a fixed permission probability scheme. This is reflected in the performance curves (Figure 3.3).

Figure 3.5 presents the optimal values for the permission probability for the constant permission probability scheme. The optimal permission probability decreases as a function of M. This could be attributed to an increase in the expected number of contending terminals.

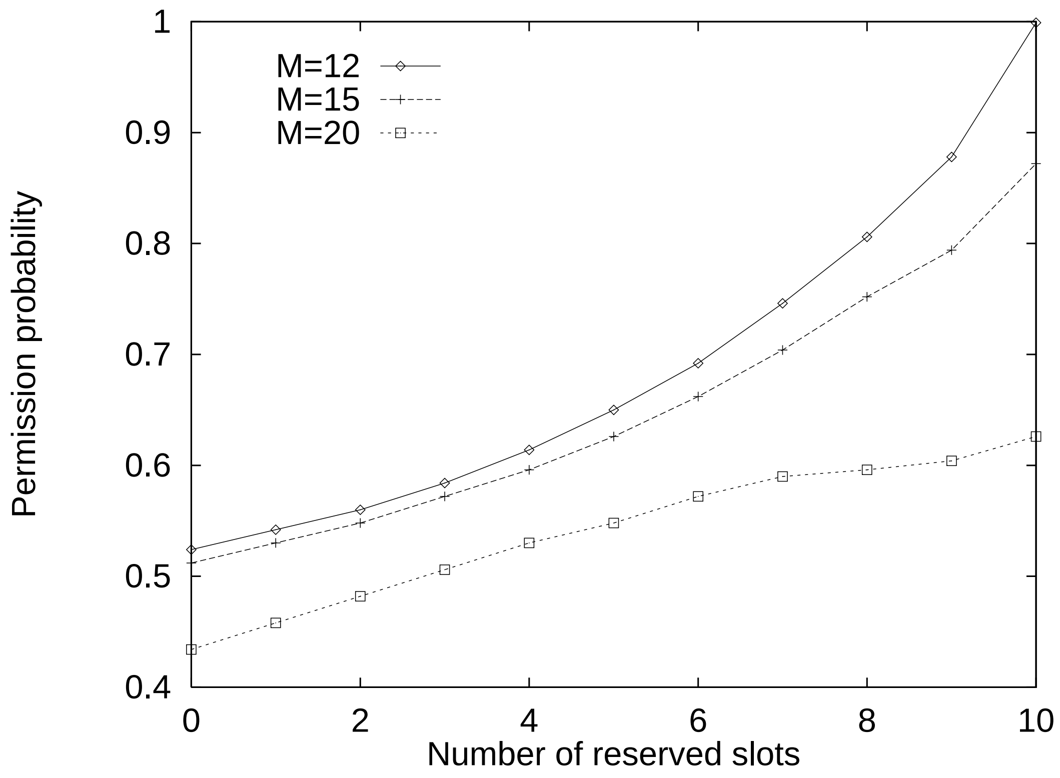


Figure 3.4: Optimal solutions for the adaptive scheme.

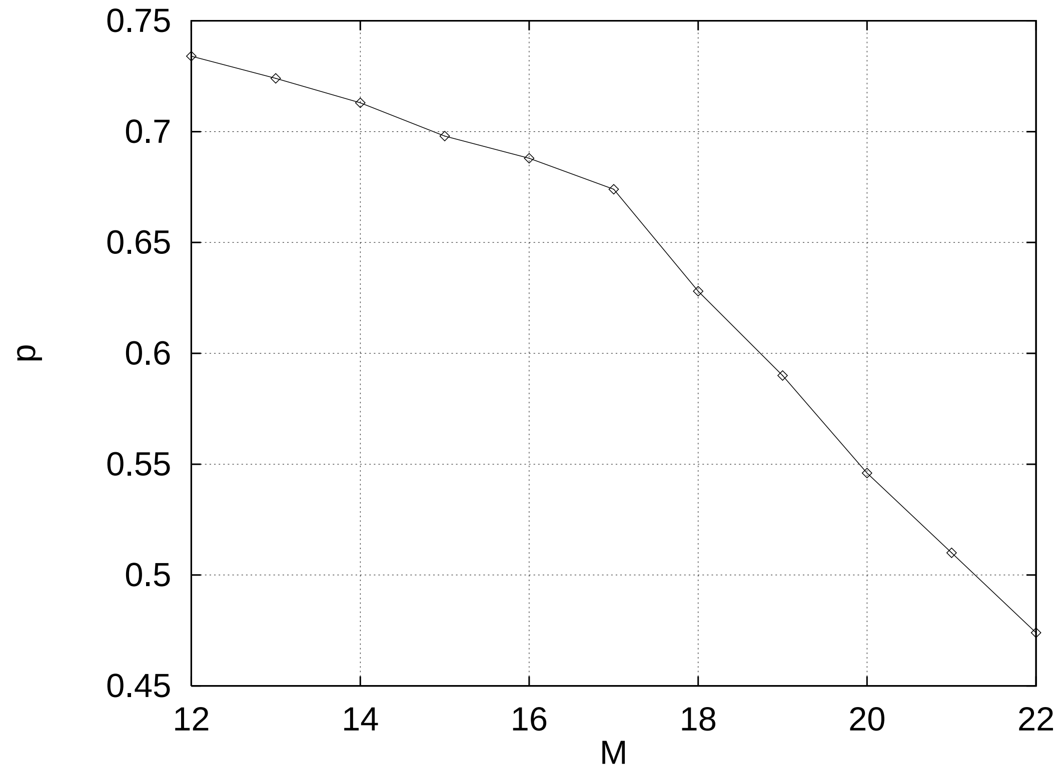


Figure 3.5: Optimal permission probability for the constant permission probability scheme.

Chapter 4

Integration of voice and data in FRMA

4.1 Introduction

In chapter 3, we introduced Frame Reservation Multiple Access (FRMA) and presented an exact analysis of the performance (voice packet dropping probability) of a voice–only system. But the next generation of personal communication networks (PCN) is expected to support a wide variety of sources (voice, video, and data are the most often quoted examples). Each of these sources will also have different QoS requirements. For example, voice packets are delay–sensitive, but can tolerate some packet loss in the event of congestion. On the other hand, most data applications (file transfer, for example) can tolerate delay, but require that all packets be transmitted without loss. Thus, any medium access control protocol designed for the next generation of PCNs should be able to accommodate the wide variety of input sources while satisfying their QoS requirements.

In this chapter, we propose a new scheme for the integration of voice and data sources in FRMA. The data sources considered here are low bit–rate sources that can tolerate packet delays of the order of 100 ms. Interactive *telnet* sessions generate data packets at a low bit–rate. Even voice terminals described in the voice–only FRMA system, from time to time, generate control packets for wireless control functions like power control and handoffs.

The FRMA protocol, described in chapter 3, will be modified to support data sources of the kind mentioned above. The basic idea in the integration of voice and data is to raise the voice packet dropping probability faced by the voice users (the QoS parameter of interest for the voice users in their packet dropping probability) to the acceptable level of 0.01, thereby freeing up a portion of the bandwidth for data sources. No attempt will made to specify any particular protocol for the data subsystem, but the amount of bandwidth that is usable for the data subsystem will be maximized. It is our belief that whatever the operating parameters of the data subsystem are, the higher the bandwidth made available to it the better its performance will be.

Previous studies [38, 62, 63, 28] on the integration of voice and data in PRMA have let the voice terminals and the data terminals contend for all the free slots. Nanda [38] presented an analysis of a PRMA system with integrated voice and data transmission using equilibrium point analysis. Each slot in a frame was classified either as reserved by a voice terminal or as an available slot. A contending voice terminal could contend for every available slot with voice permission probability p_v and, upon successful contention, obtain a reservation for a time slot in every succeeding frame until the end of its talkspurt. A data terminal could also contend for every available slot with data permission probability p_d. But a data terminal that contended successfully did not obtain a reservation for that time slot. Instead, if a data terminal had more packets to transmit, it returned to the contending state. Otherwise, the data terminal departed from the system. In order to give the voice users priority over delay insensitive data users, p_d was chosen to be less than p_v. The system was modeled using two Markov Chains corresponding to the voice and data transmission subsystems respectively. Catastrophe theory was then used to find safe regions of operation to avoid instability. This study reported a decrease in the voice capacity of the system with the addition of data traffic.

Wong and Goodman [62] propose a variant of this scheme where the data sources generate bursts of one or more data packets. Data packets that contend successfully also reserve the next $R_d = \min(B, K)$ free slots to transmit their burst, where B is the length of their burst in packets, and K is a limit enforced by the system in order to prevent long bursts from degrading the performance of the other users. This improves the performance of the data users and the throughput of the system compared to Nanda's scheme described above.

Wu *et al.*[63] present a Markov analysis of Nanda's system [38]. To make the problem tractable, they assumed that state transitions occur between the end of one frame and the beginning of the next. For the embedded Markov Chain describing this system, they derived the transition probabilities using an iterative technique. Jangi and Merakos [28] present another analytical method to analyze Reservation Random Access (RRA) protocols for wireless information networks which includes PRMA, among other RRA protocols. They have assumed that the transitions from silence to talkspurts and vice–versa occur at the frame boundaries and present a general Markov analysis of RRA protocols. Also, an iterative computational procedure based on matrix decomposition techniques is described to overcome high computational costs for large systems. They also present results of an integrated system similar to Nanda's [38] where the data users contend along with the voice users but with a different permission probability.

Our approach to the integration of voice and data sources in an FRMA system allocates additional bandwidth for data users without sacrificing the capacity of the system to support voice users. For voice users, the QoS requirement of a packet dropping probability of at most 1% for the voice subsystem is maintained. In this protocol, the contention process of the voice terminals is independent of the data traffic. So any instability in the data subsystem does not affect the voice traffic. Priority to voice terminals is implicit but limited to the extent that the quality of service (QoS) requirements of the voice traffic are satisfied. Also, this protocol does not sacrifice an important feature of PRMA – its simplicity.

The rest of the chapter is divided as follows. Section 4.2 describes the new scheme in detail along with the voice source model. Section 4.3 provides the analysis of the performance of the voice terminals. In Section 4.4, an optimization problem for the dynamic bandwidth partition is formulated. A solution to this problem would maximize the bandwidth allocated for the data terminals subject to keeping the voice packet dropping probability to at most 1%. Numerical results on the optimization of bandwidth allocations are provided in Section 4.5. In Section 4.6, a simple data subsystem is embedded in the data users' partition. Simulation is used to compare performance with results in [38, 62].

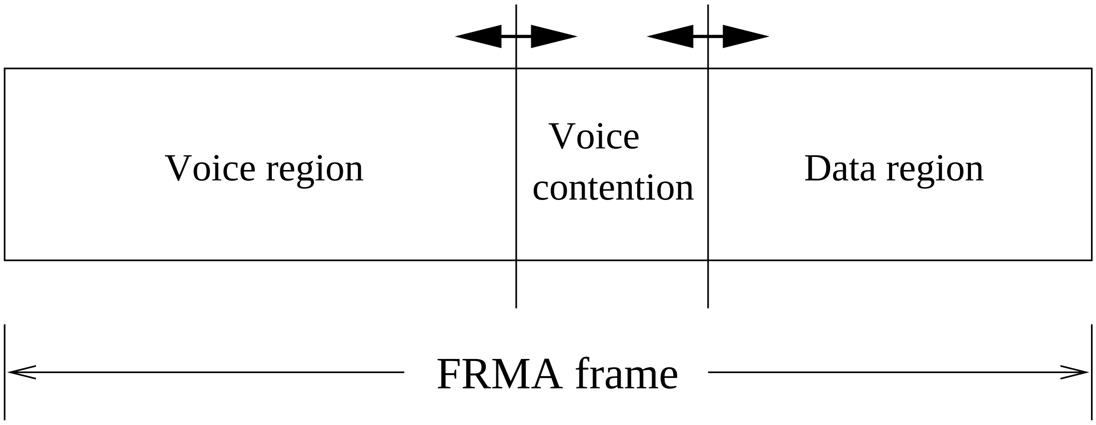


Figure 4.1: Structure of a frame in Integrated FRMA

4.2 Description of the integrated FRMA system

As in chapter 3, the input to a voice terminal follows an alternating pattern of talkspurts and silence periods, as classified by a speech activity detector. A two–state Markov process [6], as depicted in Figure 3.2, is used as the speech model. The transition probabilities are given by equations (3.2) and (3.3).

For the voice–data FRMA protocol, the entire bandwidth available for the system will be divided into three regions, namely, the voice reservation region, the voice contention region, and the data region (Figure 4.1). The boundaries between these regions will be moved, dynamically, from frame to frame, in order to keep the voice packet dropping probability below 1%. There are no restrictions imposed either on characteristics of the data sources or the protocol that the data users should be using. Whatever the nature of the data subsystem, we believe the greater the bandwidth allocated to the data subsystem, the better its performance would be. The protocol for the FRMA system that supports both voice and data sources is described below.

- At the beginning of every frame, the base classifies every time slot as either
  - reserved by a voice terminal in which case the identity of the voice terminal that holds a reservation for this time slot is also broadcasted, or
  - available for voice contention in which case all voice terminals that are in a

talkspurt but do not have a reserved slot yet can contend for these slots, or

  - a data slot for use of the data traffic.

- A voice terminal holding a reserved slot in a frame continues to hold its reservation by transmitting a packet in its reserved time slot. If the talkspurt ended in the previous frame, the terminal releases its reservation by not transmitting in its reserved slot.

- A voice terminal without a reservation, but with a packet to transmit is said to be in the contending state. Each contending voice terminal independently chooses to contend in each voice contention slot with a permission probability p that is broadcast by the base station once every frame. If exactly one terminal transmits a packet, then this contention is termed successful. The successful contending terminal is given a reserved time slot (if such a slot is available) for its exclusive use until the end of its talkspurt.

- If a voice terminal successfully contends for more than one voice contention slot in the frame, the base reserves one of the time slots for this terminal in an arbitrary manner. Thus, at the end of a frame, each voice terminal can hold a reservation for at most one time slot.

Voice terminals in the silent state that begin a talkspurt in the middle of a frame wait until the beginning of the next frame before contending for a reservation. A contending voice terminal that fails to obtain a reservation in a frame drops the oldest packet from its buffer. That is, a speech packet that is not transmitted successfully in a frame is dropped at the end of the frame. Data terminals do not get access to either the reserved slots or the voice contention slots. That is, we assume that an appropriate protocol for data users will be implemented using the remaining slots in each frame. Finally, we assume that there is no capture phenomenon, that the channel is error–free, and that transmission delays can be neglected.

4.3 Analysis of the voice subsystem

The analysis of the voice subsystem is a generalization of the analysis of the voice–only system presented in chapter 3. In the voice–only system, a slot that is not reserved by any voice terminal is available for contention. Hence, if the number of reserved slots at the beginning of a frame is R, then the number of voice contention slots is $N - R$. By comparison, in the integrated voice–data system, we control the number of voice contention slots from frame to frame to accommodate slots for the data users while maintaining the voice packet dropping probability below 1%.

Voice calls are assumed to arrive into the system in a Poisson manner with an arrival rate of λ_v calls per second. The call holding times are distributed as i.i.d. exponential random variables with a mean call duration of $1/\mu_v$ seconds. We assume that the frame duration is so short compared to the inter–arrival time of voice calls that there can be at most one call arrival per frame. Hence, the probability that a call arrives in a given frame is

$$\gamma_v = 1 - e^{-\lambda_v T}. \tag{4.1}$$

The probability that a voice call will terminate in a given frame is

$$\sigma_v = 1 - e^{-\mu_v T}. \tag{4.2}$$

Assume that all incoming calls are blocked if the number of voice calls exceed $M_{0.01}$ defined as the maximum number of voice calls that a voice–only FRMA system can support with a voice packet dropping probability below 0.01 [39].

At the beginning of a frame, let M be the number of voice calls in the system. Let R be the number of slots in the frame that are reserved by voice terminals and C denote the number of contending voice terminals. Note that the former quantity is not the same as the number of voice terminals that are in a talkspurt since there may be some terminals that are in a talkspurt but do not yet have a reserved slot for their use. Let the total number of slots per frame be denoted by N, which is a constant.

Ideally, at the beginning of every frame, the number of voice contention slots and the permission probability for the contending voice terminals should be determined as a function of the system state[1]. But at the beginning of every frame, C, the number of contending voice terminals is unknown. Hence, given the observable system state variables R and M at the beginning of each frame, the base station must choose the number of voice contention slots,

$$v = \mathbf{v}(R, M) \tag{4.3}$$

and the permission probability,

$$p = \mathbf{p}(R, M) \tag{4.4}$$

that the contending voice terminals will use in the frame. We will use a pair of vectors $(\mathbf{v}, \mathbf{p})$ to denote a voice contention policy that specifies the number of contention slots v and permission probability p for every R and M.

We now show that the voice subsystem can be modeled as a Markov chain with the state space described by (R, C, M) where at the beginning of a frame, R represents the number of slots reserved for voice terminals, C represents the number of voice terminals in the contending state, and M represents the number of voice calls in the system. We observe that a feasible state (R, C, M) must satisfy

$$\begin{array}{rcl} 0 \leq & R & \leq \min(M, N) \\ 0 \leq & C & \leq M - R \\ 0 \leq & M & \leq M_{0.01}. \end{array} \tag{4.5}$$

We will use S to denote the set of feasible states (R, C, M).

We assume that state transitions occur between the end of one frame and the beginning of the next. At the beginning of the n-th frame, let $R^{(n)}$ be the number of slots reserved for voice terminals, $C^{(n)}$ be the number of contending voice terminals, and $M^{(n)}$ be the number of voice calls in the system. Then,

$$\begin{array}{rcl} R^{(n+1)} & = & R^{(n)} + N_{CR}^{(n)} - N_{RI}^{(n)} \\ C^{(n+1)} & = & C^{(n)} - N_{CR}^{(n)} - N_{CI}^{(n)} + N_{IC}^{(n)} \\ M^{(n+1)} & = & \min(M^{(n)} + A^{(n)} - D^{(n)}, M_{0.01}) \end{array} \tag{4.6}$$

[1]As will be shown in the next paragraph, the system state can be described by (R, C, M).

where:

$N_{CR}^{(n)}$ is the number of contending voice terminals that contend successfully and hence obtain a reservation at the end of the n-th frame,

$N_{RI}^{(n)}$ is the number of voice terminals with reservations that complete their talkspurts during the n-th frame and hence move to the idle (SIL) state. This random variable is distributed as Binomial$(R^{(n)}, \gamma)$,

$N_{CI}^{(n)}$ is the number of contending voice terminals that complete their talkspurts during the n-th frame and move to the idle (SIL) state. This random variable is distributed as Binomial$(C^{(n)} - N_{CR}^{(n)}, \gamma)$,

$N_{IC}^{(n)}$ is the number of silent voice terminals that begin a talkspurt during the n-th frame and move to the contending state. This random variable is distributed as Binomial$(M^{(n)} - R^{(n)} - C^{(n)}, \sigma)$,

$A^{(n)}$ is the number of new voice calls arriving into the system during the n-th frame. We assume that the duration of a frame is so short compared to the duration of the inter arrival time of voice calls there can be at most one arrival of a voice call in a frame. So this random variable is a Bernoulli trial with parameter γ_v.

$D^{(n)}$ is the number of voice calls that complete during the n-th frame and depart from the system. This random variable is distributed as Binomial$(M^{(n)} - R^{(n)} - C^{(n)}, \sigma_v)$

where γ and σ are given in equations (3.2) and (3.3). From equation (4.6), we observe that $(R^{(n)}, C^{(n)}, M^{(n)})$ is a Markov state description. In addition, for any finite number of voice users, the system has a finite number of states. Since delayed voice packets are dropped, any choice of $(\mathbf{v}, \mathbf{p})$, such that $0 \leq p_i \leq 1$, $0 \leq v_i \leq N - i$, $i = 0, 1, \ldots, \min(M, N)$, would lead to the state $(0, 0, 0)$ being reachable from every other state. Thus, the Markov chain has a single communicating class and a unique stationary distribution $\pi(R, C, M)$ for any choice of $(\mathbf{v}, \mathbf{p})$.

In order to compute the transition probability matrix of the Markov Chain, we must derive the conditional probability distribution of N_{CR}, the number of contending voice terminals that succeed in obtaining a reservation at the end of a frame. Suppose the system is in state (R, C, M) at the beginning of an arbitrary frame. The number of voice contention slots is given by $v = \mathbf{v}(R, M)$ and the permission probability is $p = \mathbf{p}(R, M)$. Let J denote the number of successful transmissions by contending voice terminals during the frame. A successful transmission by a contending voice terminal occurs in a voice contention slot with probability

$$q = Cp(1-p)^{C-1}, \qquad C > 0, \tag{4.7}$$

and a failure (collision or idle) occurs with probability $(1-q)$. Since the outcome in one voice contention slot is independent of the outcomes in all other voice contention slots, the number of successful transmissions, J, has a binomial distribution.

$$P\{J = j\} = \begin{cases} \binom{v}{j} q^j (1-q)^{v-j}, & j = 0, 1, \ldots, v, \\ 0, & \text{otherwise.} \end{cases} \tag{4.8}$$

Since a contending voice terminal may have more than one successful transmission in a frame, N_{CR}, the number of contending terminals that succeed in making reservations, may be less than J, the number of successful transmissions. The probability distribution of N_{CR} can be expressed as

$$P\{N_{CR} = s\} = \sum_{j=s}^{v} P\{N_{CR} = s | J = j\} P\{J = j\}. \tag{4.9}$$

We derive the expression for $P\{N_{CR} = s | J = j\}$ in Appendix A.

From the transition probability matrix derived above, we can derive the steady–state probabilities, $\pi(R, C, M)$ of the states of the Markov Chain. Speech packets that are not transmitted successfully in a frame are dropped at the end of the frame. The long term packet dropping probability is computed as the ratio of the average number of speech packets dropped in a frame to the average number of packets generated per frame.

Since each contending voice terminal that fails to obtain a reservation at the end of a frame drops a packet, the number of dropped packets in a frame equals the number of unsuccessful contending voice terminals. Each terminal that holds a reservation at the beginning of a frame successfully transmitted a packet in the previous frame since any terminal with a successful transmission in a frame continues to hold the reservation in the next frame. Hence R is the number of successful packets in the previous frame and if D is the number of packets that are dropped, the sum $R + D$ must equal the number of packets generated during the previous frame. Hence, the packet dropping probability is given by

$$p_{\text{drop}} = \frac{E[D]}{E[R+D]}. \tag{4.10}$$

A contending voice terminal fails to make a reservation in an arbitrary voice contention slot with probability $1 - p(1-p)^{C-1}$. A contending voice terminal drops a packet at the end of a frame if it is not successfully transmitted in any of the v voice contention slots. Since the outcomes in the v voice contention slots are independent, an arbitrary contending voice terminal drops a packet with probability

$$\delta(R, C, M) = (1 - p(1-p)^{C-1})^v \tag{4.11}$$

with v and p chosen according to equations (4.3) and (4.4) respectively. Summing over all the C contending terminals, the expected number of dropped packets becomes

$$E[D|R, C, M] = C\delta(R, C, M). \tag{4.12}$$

Averaging over the stationary distribution of the system, the expected number of dropped packets per frame is

$$E[D] = \sum_{(R,C,M)\in S} E[D|R, C, M]\pi(R, C, M). \tag{4.13}$$

The expected number of voice packets generated per frame, $E[R + D]$, depends only on the probability distribution of the number of voice users in the system and the speech activity, α. So

$$E[R+D] = \sum_{m=1}^{M_{0.01}} m\alpha P\{M = m\} = E[M]\alpha. \tag{4.14}$$

Since voice calls arrive as a Poisson process of rate μ_v and calls are blocked only when the there are $M_{0.01}$ voice calls in the system, the number of voice calls in the system has the truncated Erlang distribution

$$P\{M = m\} = \frac{(\lambda_v/\mu_v)^m/m!}{\sum_{n=0}^{M_{0.01}} (\lambda_v/\mu_v)^n/n!} \quad (0 \le m \le M_{0.01}). \tag{4.15}$$

Therefore, the packet dropping probability, is given by

$$p_{\text{drop}} = \frac{1}{E[M]\alpha} \sum_{(R,C,M)\in S} E[D|(R,C,M)]\pi(R,C,M). \tag{4.16}$$

4.4 Selection of Design Parameters

A pair of vectors, $(\mathbf{v}, \mathbf{p})$, is said to be *feasible* for the voice subsystem if it satisfies the criterion that no more than 1% of the voice packets arriving into the system will be dropped. In choosing "good" values for the design parameters, we will restrict ourselves to the set of feasible solutions. This will guarantee that the QoS requirements of the voice terminals are satisfied. While attempting to satisfy the QoS requirements of the voice terminals, we would also like to get the "best" performance possible for the data users. At this time, we do not know what the operating parameters of the data subsystem (like the arrival process, the probability distribution of the service demand, the multi–access protocol that the data users will use, etc.). But we assume that whatever the operating parameters of the data subsystem, the performance of the data subsystem would be better if more bandwidth is allocated to it.

If the above assumption is true, then a pair of vectors, $(\mathbf{v}, \mathbf{p})$, is said to be *optimal* if it is feasible for the voice subsystem, and it maximizes the expected bandwidth available for the data subsystem. The expected number of slots per frame allocated to the data users is given by

$$E[N - R - v(R,M)] = \sum_{(R,C,M)\in S} (N - R - v(R,M))\pi(R,C,M). \tag{4.17}$$

So the problem of finding optimal values for the design parameters can be formulated as

$$\begin{aligned} \text{maximize} \quad & E[N - R - v(R,M)] \\ \text{subject to} \quad & p_{\text{drop}} \le 0.01 \\ & 0 \le p(R,M) \le 1 \\ & 0 \le v(R,M) \le N - R. \end{aligned} \tag{4.18}$$

In (4.18), $v(R,M)$ is shown to be a continuous variable. However, we must allocate an integral number of voice contention slots in any frame. This relaxation of the integer constraint on $v(R,M)$ can be handled as follows. When a solution to (4.18) specifies a non-integral value for $v(R,M)$ in a frame with M voice calls and R reserved slots, we choose an integer–valued random number $V(R,M)$ of voice contention slots such that $E[V(R,M)] = v(R,M)$. In particular, we define

$$f(R,M) = v(R,M) - \lfloor v(R,M) \rfloor \tag{4.19}$$

so that

$$V(R,M) = \begin{cases} \lfloor v(R,M) \rfloor & \text{with probability } 1 - f(R,M) \\ \lceil v(R,M) \rceil & \text{with probability } f(R,M) \end{cases} . \tag{4.20}$$

The transition probability matrix of the system, in particular equation (4.9), has to be modified to account for the randomization in (4.20). Hence, the optimization problem becomes a non-linear programming problem in the continuous parameters $v(R,M)$ and $p(R,M)$ rather than an integer non-linear programming problem.

It is clear that if we reduce the number of voice contention slots, the mean access delay increases and hence voice packet dropping probability increases. Given any $(\mathbf{v},\mathbf{p})$ such that $p_{\text{drop}} < 0.01$, the optimization procedure decreases the number of voice contention slots appropriately to increase the number of data slots. This procedure increases voice packet dropping probability and so we stop when $p_{\text{drop}} = 0.01$.

We know that $E[R]$ is the expected number of successfully transmitted voice packets in a frame. The expected number of voice packets generated by all voice terminals is $E[M]\alpha$. If exactly 1% of all voice packets are dropped, then, at optimum,

$$E[R] = 0.99E[M]\alpha, \tag{4.21}$$

which is independent of the choice of $(\mathbf{v},\mathbf{p})$. So the optimization problem (4.18) can be modified as

$$\begin{aligned} \text{minimize} \quad & E[v(R,M)] \\ \text{subject to} \quad & p_{\text{drop}} \leq 0.01 \\ & 0 \leq p(R,M) \leq 1 \\ & 0 \leq v(R,M) \leq N - R. \end{aligned} \tag{4.22}$$

The decision variables in the above optimization problems are $(\mathbf{v},\mathbf{p})$. The functions $E[v(R,M)]$ and p_{drop} are non–linear in the decision variables and do not have a closed form due to the presence of the stationary probabilities $\pi(R, C, M)$. This makes it difficult to solve the above problem using non–linear programming techniques that use derivatives. Computing the values of these two functions for a particular $(\mathbf{v}, \mathbf{p})$ involves the inversion of a transition probability matrix of the Markov Chain to get the steady–state probabilities. Moreover, computation of the transition probability matrix is non–trivial.

The number of states in this Markov Chain is typically of the order of $O(NM_{0.01}^2)$. In order to keep the size of the optimization problem computationally small, we will assume the number of voice terminals in the system, M, is constant and solve for the "optimal" values for the design parameters for that value of M. For a constant number M of voice terminals, the number of states in the Markov Chain is $O(NM)$. We perform this optimization for each value of M such that $0 \leq M \leq M_{0.01}$. We then use these "optimal" solutions to compute exactly the performance of the voice subsystem with varying M due to call arrivals and departures. It is important to note that, although the optimization problem is difficult to solve, it needs to be solved only once, and that can be done off–line, as part of the system design.

These heuristic techniques have yielded approximate solutions to (4.22) that we believe are close to optimum. Even though the solutions to the optimization problem (4.22) is approximate, it should be noted that the corresponding performance measures (voice packet dropping probability and the expected number of data slots per frame) are computed exactly from the resulting Markov chain. Thus, the numerical results that are presented in the next section are exact, but are not necessarily optimal.

Properties of the optimal voice contention policy permit us to calculate or estimate quickly the impact of an increase in the system bandwidth. Let $(\mathbf{v}^*, \mathbf{p}^*)_{M,N}$ denote the optimal solution for the non–linear programming problem described in (4.22) with the number of slots per frame being N and the number of voice terminals being M. Then the following observations are true.

- Suppose M_0 and N_0 are such that $M_0 \leq N_0$. Then, the voice contention policy $(\mathbf{v}^*, \mathbf{p}^*)_{M_0,N_0}$ is optimal for any $N \geq M_0$, because the state space of the Markov Chain remains the same and the transition probability matrix depends only on

Source rate	32 Kbps
Channel rate	432 Kbps
Overhead	80 bits
Frame duration	11 ms
Number of slots per frame	11
Average talkspurt duration	1 sec
Average silence duration	1.35 sec

Table 4.1: Parameters of an 11 slot FRMA System

the voice activity parameters and the decision variables.

- Let $D^*(M_0, N_0)$ denote the maximum available data bandwidth with M_0 voice terminals and N_0 slots per frame under the optimal policy $(\mathbf{v}^*, \mathbf{p}^*)_{M_0,N_0}$. For a system in which the number of slots is $N > N_0$, $(\mathbf{v}^*, \mathbf{p}^*)_{M_0,N_0}$ would restrict the voice users to a subset of N_0 slots while providing $p_{\text{drop}} \leq 0.01$. For the N slot system with voice contention policy $(\mathbf{v}^*, \mathbf{p}^*)_{M_0,N_0}$, an additional $N - N_0$ slots per frame would be available for data. Hence, under the optimal policy for a system with M_0 voice terminals and $N \geq N_0$ slots,

$$D^*(M_0, N) \geq D^*(M_0, N_0) + N - N_0. \tag{4.23}$$

Thus, any increase in the total available bandwidth translates to additional data bandwidth given the number of voice terminals does not change.

4.5 Bandwidth Allocation – Numerical Results

It is instructive to examine the bandwidth allocations as a function of the number of voice terminals M in the system. In particular, for a system with $N = 11$ slots described by Table 4.1, we have found approximate optimal solutions for (4.22). Figure 4.2 plots the proportion of the available bandwidth that is allocated for each of the three classes – voice traffic, voice contention, and data, as a function of M. The voice packet dropping probability is exactly equal to 1% whenever there was at least one voice terminal in the system. From [42], we know that $M_{0.01} = 18$. From the figure, we can observe that even when the voice subsystem is operating at its capacity, we can have a maximum data

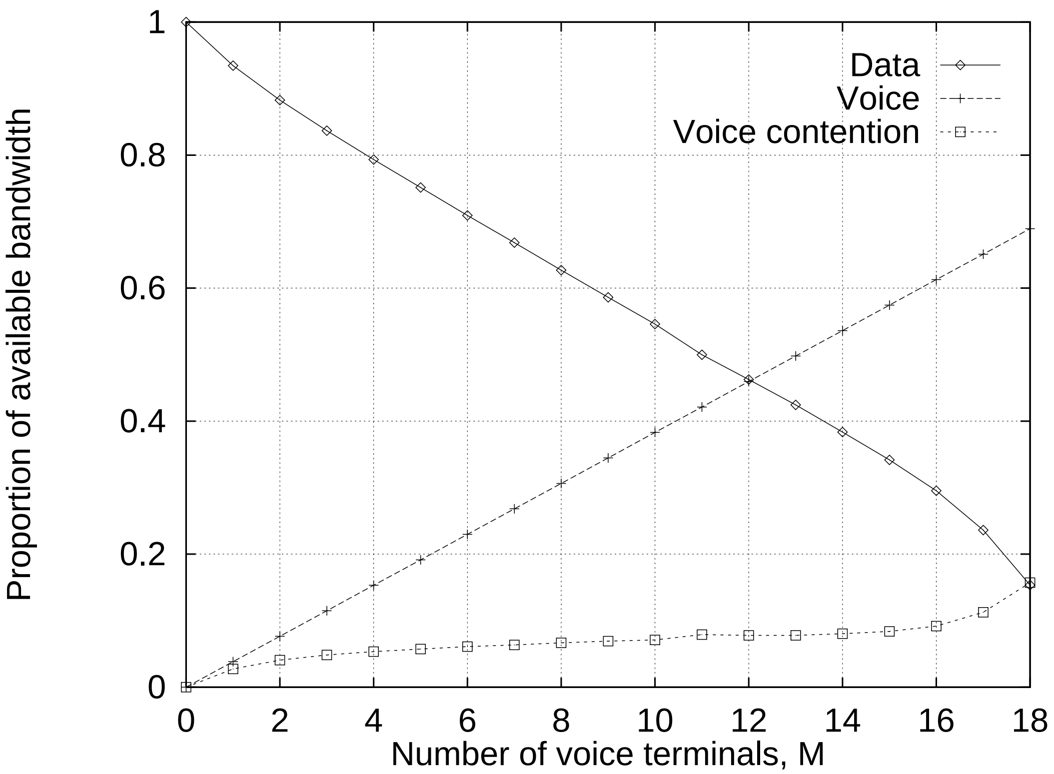


Figure 4.2: The proportion of bandwidth required by each of the three classes of traffic, namely, voice traffic, voice contention, and data as a function of the number of voice terminals. The number of voice terminals in the system, at each point in the graph, was assumed to be a constant. Parameters for this 11 slot system are given in Table 4.1.

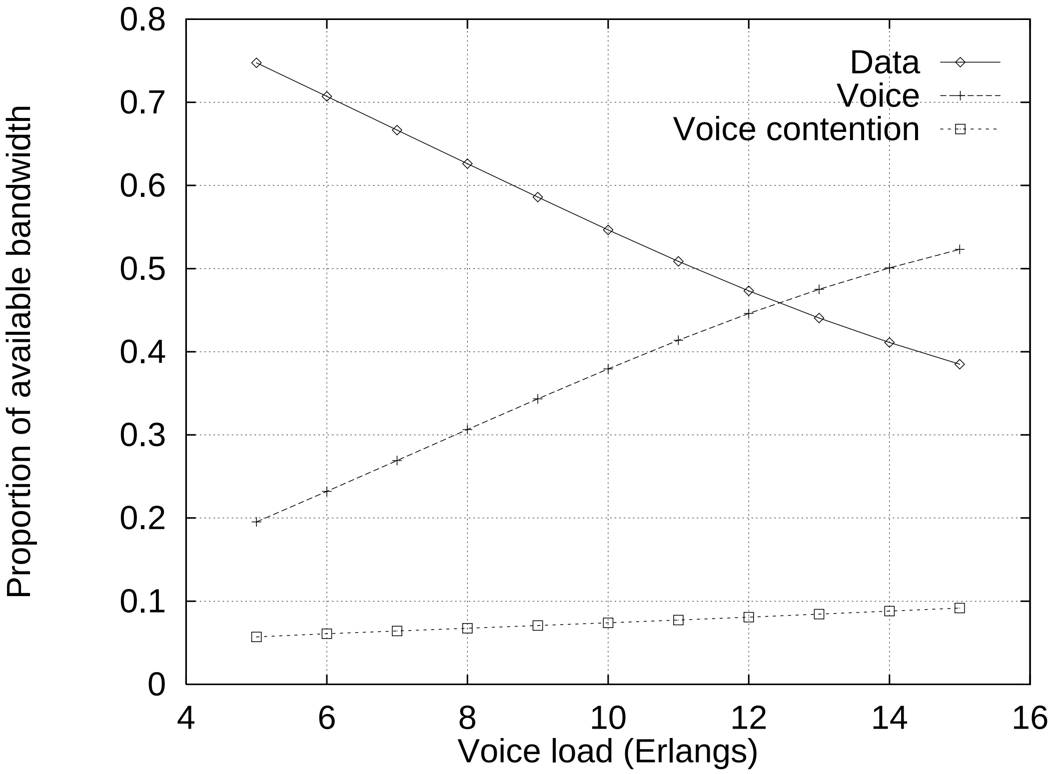


Figure 4.3: The proportion of available bandwidth required by each of the three classes of traffic as a function of the voice load on the system. Parameters for this 11 slot system are given in Table 4.1.

throughput of about 17%. This is due to the fact that in the voice–only system, when the number of terminals is constant at $M_{0.01}$, the voice packet dropping probability is around 0.0088 [42]. Since this is lower than the bound of 0.01, we are able to push the voice packet dropping to 0.01 and accommodate some data traffic.

Figure 4.3 plots, as a function of voice load, the proportion of bandwidth allocated to each of the three classes of traffic for the 11 slot system with a Poisson call arrival process. With a voice call blocking probability of 0.02, we can support about 11 Erlangs of voice traffic and about 50% of the channel can be utilized by the data users.

Figure 4.4 shows the optimal values for the permission probability as a function of R. These values have been plotted for three different values of M, the number of voice terminals in the system. The permission probability values seem to be an increasing function of R (with M being constant), and a decreasing function of M (for the same values of R). The optimal values of the number of voice contention slots as a function of R are plotted in figure 4.5. These values are plotted for three different values of M. One reason for the fewer voice contention slots for very low values of R could be that these states have a lower probability. Also, the structure of these solutions are highly dependent on the heuristic used to solve them. Figure 4.6 presents the average number of data slots as a function of R.

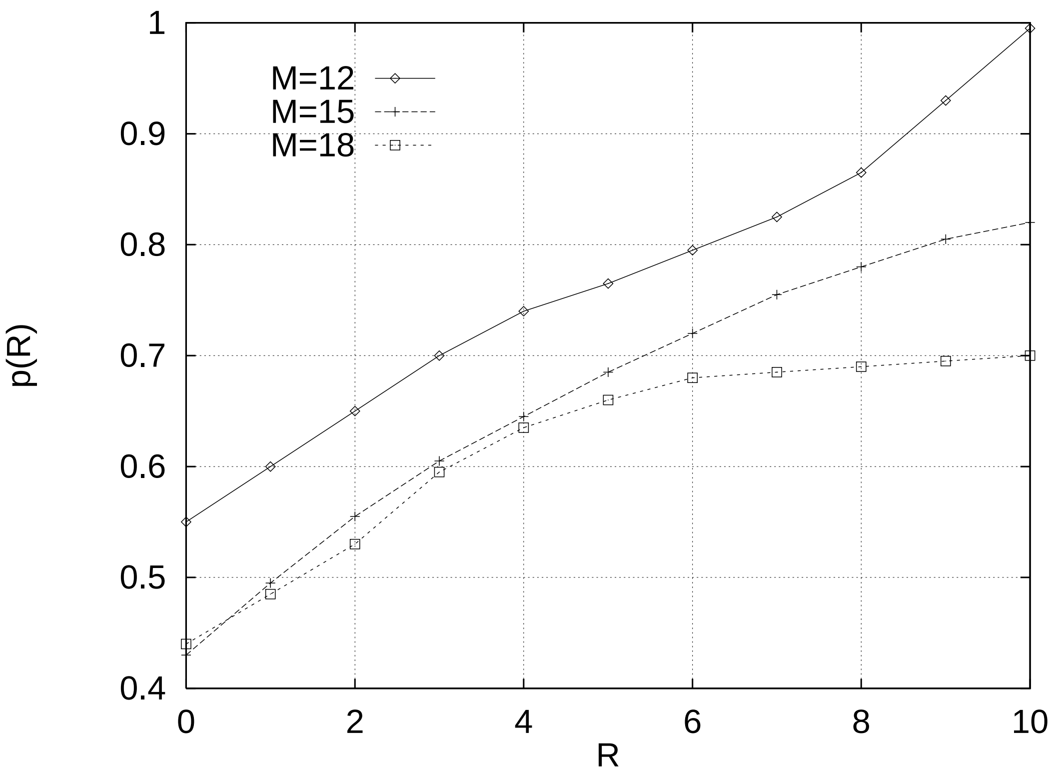


Figure 4.4: Optimal permission probability vectors for three different values of M.

In order to compare the performance of our system with Nanda [38], and Wong and Goodman [62], we also studied a 20 slot system with parameters described in Table 4.2. We found the capacity of the voice–only system to be 36 voice terminals, which is consistent with the results in in the literature [38, 62]. Thus, there is no loss in capacity due to transmitting acknowledgments once per frame instead of once every slot.

In Figure 4.7, we plot the proportion of the available bandwidth that is allocated to

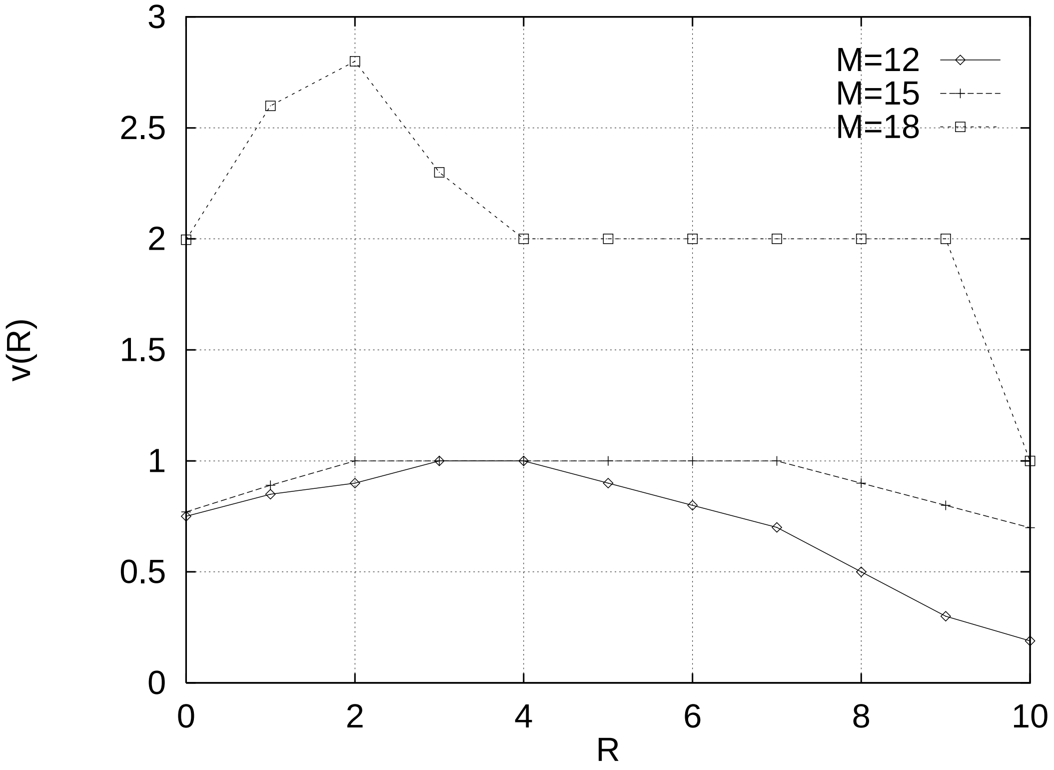


Figure 4.5: Optimal values for the number of voice contention slots as a function of R, for three different values of M.

Source rate	32 Kbps
Channel rate	720 Kbps
Overhead	64 bits
Frame duration	16 ms
Number of slots per frame, N	20
Average talkspurt duration, t_1	1 sec
Average silence duration, t_2	1.35 sec

Table 4.2: Parameters for a 20 slot FRMA System

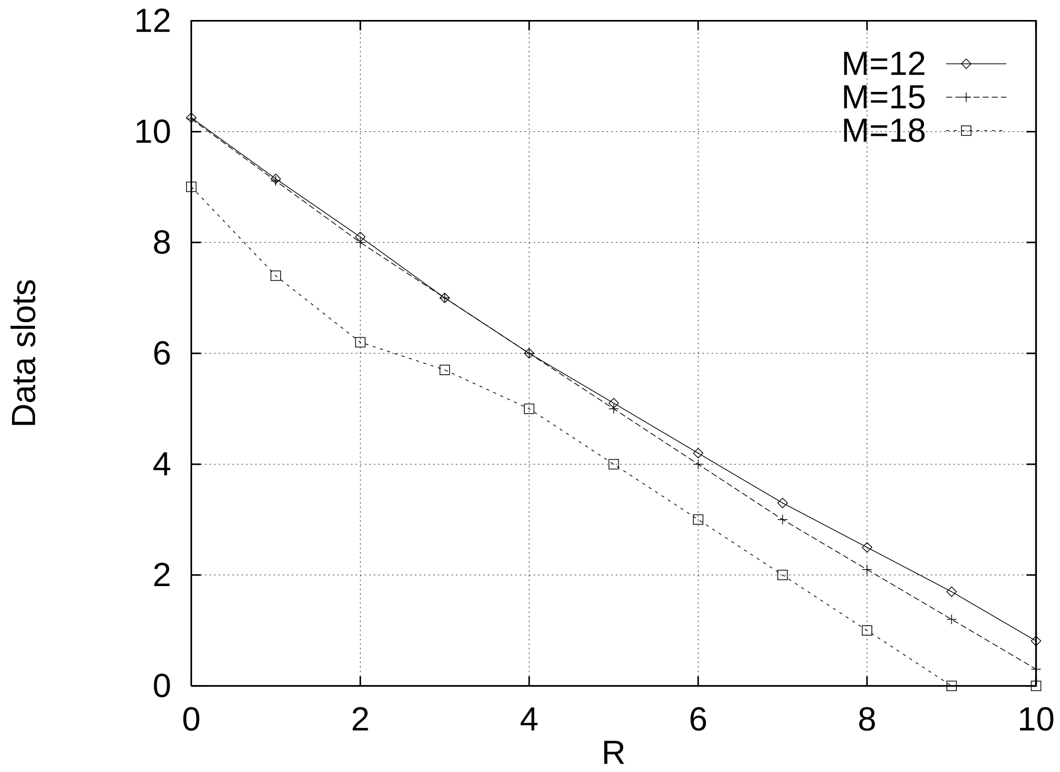


Figure 4.6: Average number of data slots as a function of R, for three different values of M.

the three classes of traffic. This figure is the analogue of the results in Figure 4.2, but for the 20–slot system. Both the 11 and 20 slot systems exhibit similar performance with regions of linear maximum data capacity followed by a sharp drop near $M = M_{0.01}$.

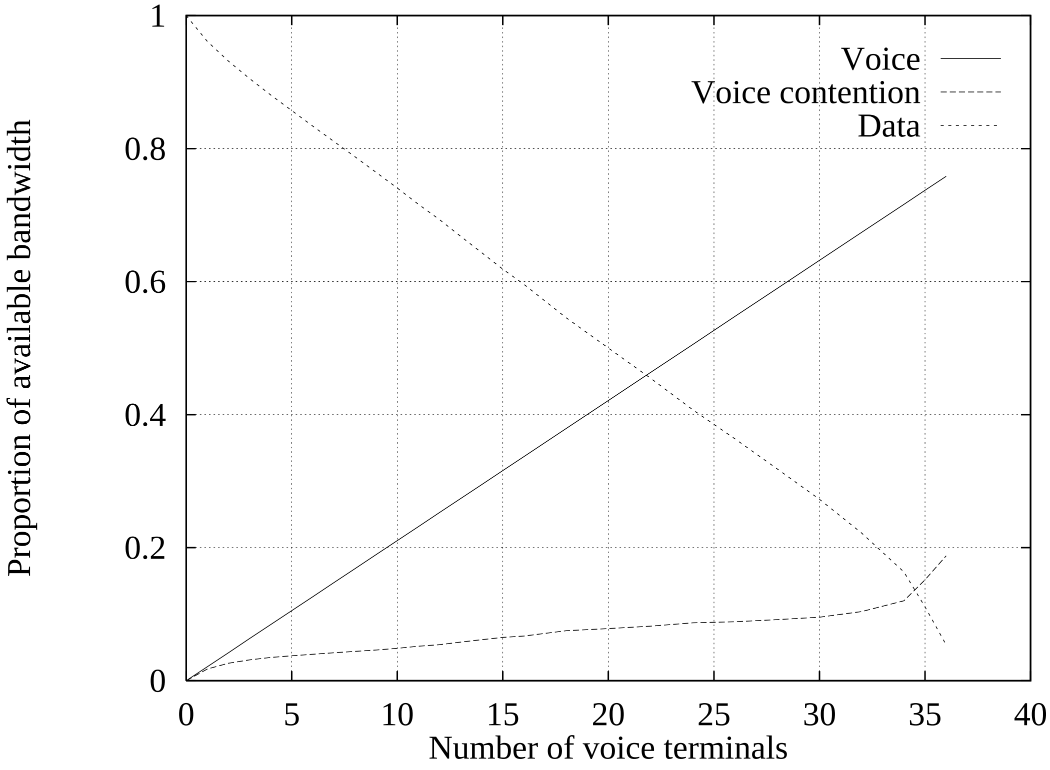


Figure 4.7: Proportion of available bandwidth allocated to each of the three classes of traffic. The number of voice terminals is constant, the decision variables for the voice subsystem were obtained from solutions to (4.22). Parameters for this 20 slot system are given in Table 4.2.

4.6 Performance of the Data Subsystem

So far, we have only considered the bandwidth allocations that result from our efforts to solve the optimization problem (4.22). We have found that significant capacity can be dynamically partitioned for the data users. In order to make comparisons with earlier work [38, 62], we will embed a simple data subsystem within the data users' partition. In this section, we evaluate by simulation the performance of the data subsystem.

For the data subsystem, the arrival of data packets is modeled as a Poisson process with a mean rate of R_d bits per second at each data terminal. The number of data terminals is fixed and any data terminal with at least one packet in its buffer is considered backlogged. All other data terminals are considered to be in the idle state. The data subsystem cycles through two phases, each consisting of a number of frames.

- a contention phase – in which all backlogged data terminals contend for channel access. They do so by choosing to contend in a frame according to a data permission probability p_d, and then choosing one of the data slots at random. The contention phase ends when at least one data terminal contends successfully in a frame.

- a service phase – in which all successful data terminals are served according to a random order of service. During this phase, the base station identifies the data terminal that could transmit in each data slot. When a particular data terminal that has been assigned a data slot fails to transmit in that slot, then the base station assumes that this data terminal has moved to the idle state and removes this terminal from the service queue. The service phase ends when the service queue is empty.

The value of p_d and the choice of service discipline have an impact on the data subsystem performance. The dynamic partitioning of the voice and data subsystems permits us to optimize the data subsystem within the data users' partition. Moreover, the choice of data subsystem would have no effect on the QoS received by the voice users. The optimization of the data subsystem, however, is beyond the scope of this work and could be the subject of a future study.

Following [38, 62], the number of voice terminals M_v, and the number of data terminals M_d, were both held constant at 20 in the simulation experiments. The buffer at each data terminal was assumed to hold 1000 data packets. As in [62], the data subsystem was assumed to be stable, as long as there were no buffer overflows at any data terminal for the duration of the simulation (200000 frames). The permission probability for the voice terminals and the size of the voice contention region were chosen from the solution to the problem (4.22) with $M = 20$.

Wong and Goodman [62] plot the average packet delay for a given data rate, R_d, at each data terminal with $M_v = 20$ and $M_d = 20$. They conclude that the maximum data rate that can be supported by their system is 3500 bps, compared to 1400 bps for Nanda's system [38]. In Figure 4.8, we plot the average packet delay for different values of R_d with $M_v = M_d = 20$ and compare the results[2] with those in Nanda [38], and Wong and Goodman [62]. Figure 4.8 shows that the sustainable data rate with the integrated FRMA is about 10,000 bps, which is roughly 7 times the sustainable data rate in Nanda [38], and thrice that in Wong and Goodman [62]. Of course, one must keep in mind that the additional data capacity is due, in part, to the fact that even when the overall voice/data system is lightly loaded, the voice users will endure 1% packet dropping.

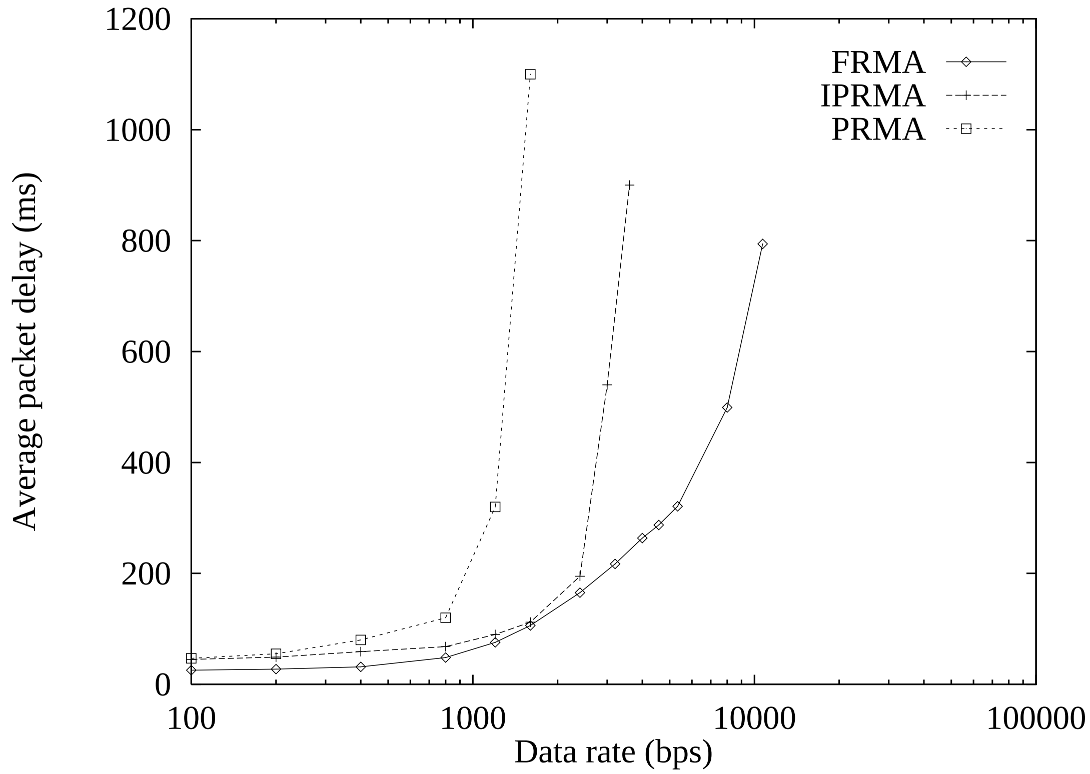


Figure 4.8: Average packet delay as a function of the data rate R_d at each data terminal. $M_v = M_d = 20$.

[2]The two curves for PRMA and IPRMA shown here were derived from Wong and Goodman [62].

4.7 Conclusions

We have proposed a new scheme for the integration of voice and data over FRMA, a variant of PRMA. The voice and data subsystems are separated logically. Voice traffic is given priority that is implicit and limited by the satisfaction of its QoS requirement. In this case, the QoS requirements for the voice traffic is that the voice packet dropping probability not exceed 1%.

The voice sub–system has been formulated as a Markov Chain and an exact analysis of its performance (voice packet dropping probability) has been carried out. An optimization problem to maximize the bandwidth allocated to the data users while satisfying the QoS requirements of the voice users has been formulated. Since this optimization problem is difficult to solve using non–linear programming solution techniques, we have adopted heuristic approaches to obtain "approximate" solutions for the decision variables. But it should be stressed that the results summarized in the previous section are exact, and we believe they are very close to optimum.

The capacity of the system to support voice users is not sacrificed in the integration of data users into the system. Also, since the voice and the data subsystems are logically separated, any instability in one will not affect the other. The numerical results indicate that the bandwidth allocated for the data users is quite significant. Overall, the new protocol does not sacrifice the simplicity of PRMA while improving the performance of the system.

In a preliminary study of the data subsystem using simulations, we found that the integrated FRMA system described above can support a substantially higher data rate with lower delays than the integrated PRMA systems of Nanda [38] and Wong and Goodman [62].

Chapter 5

Wireless ATM

5.1 Introduction

Asynchronous Transfer Mode (ATM) is emerging as the primary networking technology for the next generation, multi-media communication network known as B–ISDN[1]. B–ISDN is the worldwide digital system, currently being installed in many countries, that will permit its users to communicate over high–quality, high–speed digital communication channels.

ATM protocols are designed to handle isochronous (time critical) data such as video and telephony (audio), in addition to more conventional data communications between computers. ATM protocols are capable of providing a homogeneous network for all traffic types. The same protocols are used regardless of whether the application is to carry conventional telephony, entertainment video, or computer network traffic over local area networks (LANs), metropolitan area networks (MANs), or wide area networks (WANs).

ATM technology is based on small, constant-sized cells (53 bytes in size that includes a 48 byte payload and a 5 byte header) that permit sufficiently rapid switching that multiple isochronous data can be statistically multiplexed together, along with computer network traffic. No longer will a communications channel be limited to a fixed data rate, rather any application uses only the bandwidth required. If an application requires additional bandwidth for bursty data, it can request the additional bandwidth.

In [54, 50], an architecture for a wireless ATM system is proposed. This system

[1] Broadband Integrated Services Digital Network

is intended to provide mobile terminals with high–speed cell relay services qualitatively similar to those provided by future B–ISDN/ATM, thereby facilitating seamless support of network–based multimedia applications on both fixed and portable terminals. Wireless ATM may be viewed as a general solution for next–generation personal communication networks (PCN) which will be required to support multimedia. The proposed system is intended to be an integral part of a high–speed ATM network with unified wired and wireless services via standard ATM network and signaling / control layers, augmented to support mobility. New wireless–specific medium access control (MAC), data link control (DLC), and wireless network control layers are incorporated into the ATM protocol stack to accommodate the radio channel requirements. In order to maintain uniformity of service classes and quality–of–service (QoS) parameters, the wireless network is designed to support standard ATM service interfaces including ABR (available bit–rate), CBR (constant bit–rate), and VBR (variable bit–rate). Of course, quantitative limits on bit–rate, cell loss, delay, etc. may be imposed by the limitations of the radio medium.

The basic approach to wireless ATM, as described in [54], is to use standard ATM cell for network level functions, while a wireless header/trailer is added on the radio link for wireless channel specific protocol layers – Medium Access Control (MAC), Data Link Control (DLC), and wireless network control. Figure 5.1 shows the interface between the fixed and the wireless networks. ATM services with QOS requirements are provided on an end–to–end basis with standard ATM signaling functions being terminated at the mobile unit. The protocol layering in wireless ATM is harmonized with that of standard ATM. As shown in Figure 5.1, new wireless channel specific MAC, DLC and wireless network control layers are incorporated into the ATM protocol stack. A limited number of mobility extensions to ATM network and control layers will be required to support additional functions such as location management, handoff, QoS specification / renegotiation, etc.

In order to overcome the effects of the inherent high error rate associated with the radio channel, a new data link control (DLC) is used in wireless ATM to provide an

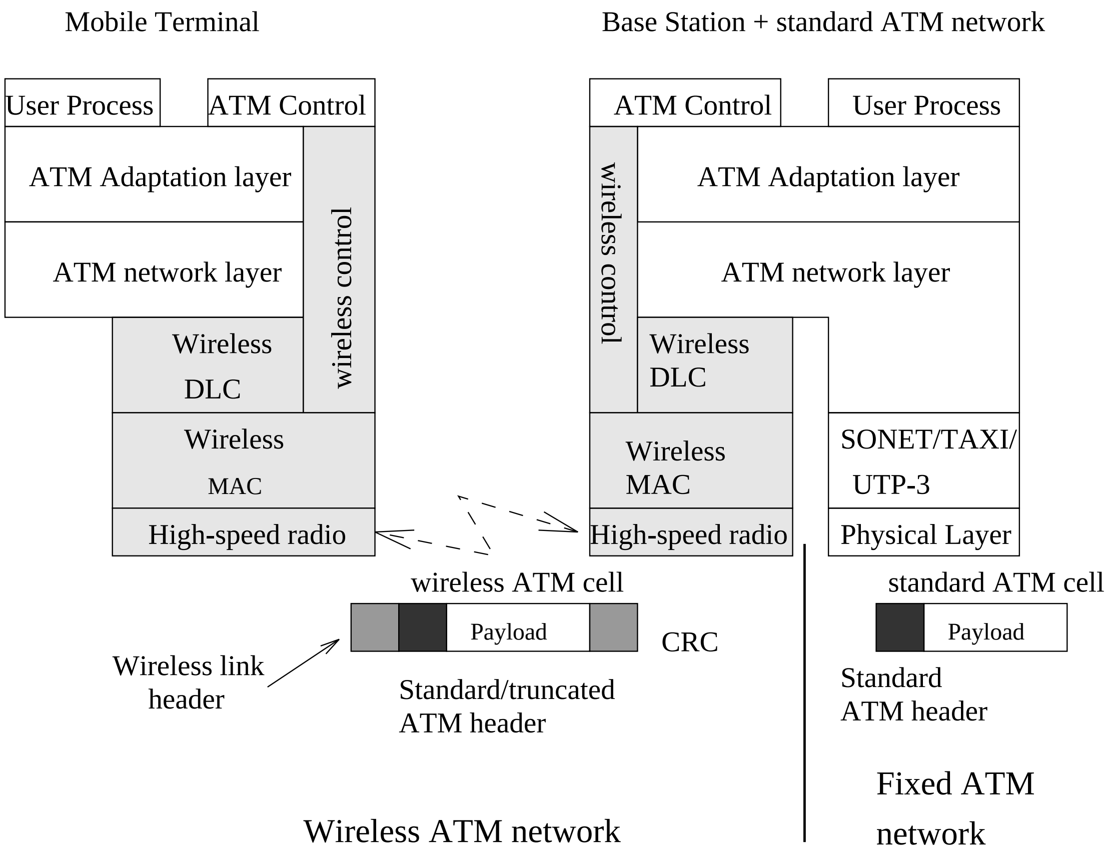


Figure 5.1: Interface between the fixed and the wireless networks.

additional layer of error protection to ATM services. DLC protocols are not only applied to packet–mode ABR services, but also to stream–mode CBR and VBR services, with the retransmission schemes matched to the requirements of the individual service classes. For example, no intrinsic time delay constraints need to be applied to error recovery procedures for ABR services, while for the CBR and VBR services, the DLC must maintain acceptable isochronous throughput and delay in the cell stream passed up to the ATM layer. The proposed protocols have been validated using a software emulator which incorporates a choice of radio channel models, dynamic TDMA/TDD MAC, and ABR or CBR DLC. Numerical results show that the data link control protocols under consideration can significantly improve wireless ATM service quality over impaired radio channels for both packet data ABR and stream type CBR virtual circuits.

In this chapter, we describe, briefly, the architecture of a wireless ATM system based on a dynamic TDD/TDMA framework [54] (Section 5.2). A more detailed description can be found in [65]. Performance evaluation results from a detailed software emulation of the DLC and the MAC are given in section 5.3. Coverage of other aspects of the wireless ATM system under consideration can be found in [52] (system considerations), [66] (mobility signaling and control), and [53] (video transport).

5.2 System Architecture

Figure 5.2 shows the architecture of the Data Link Control (DLC) and the Medium Access Control (MAC) protocol layers. Standard ATM cells arrive into the DLC layer corresponding to the particular virtual circuit (VC). The DLC replaces the standard ATM header with the wireless ATM (WATM) header and requests the Supervisory MAC (S–MAC) to allocate slots for the transmission of these cells. The S–MAC schedules transmissions/receptions for a particular frame and passes this schedule table to the Core MAC (C–MAC), which gets the cells from the corresponding DLC for transmission (or transfers the received cells to the DLC of the corresponding VC). At the receiving end, the DLC transfers received cells (after any required error recovery) to the ATM layer after replacing the WATM header with the standard ATM header.

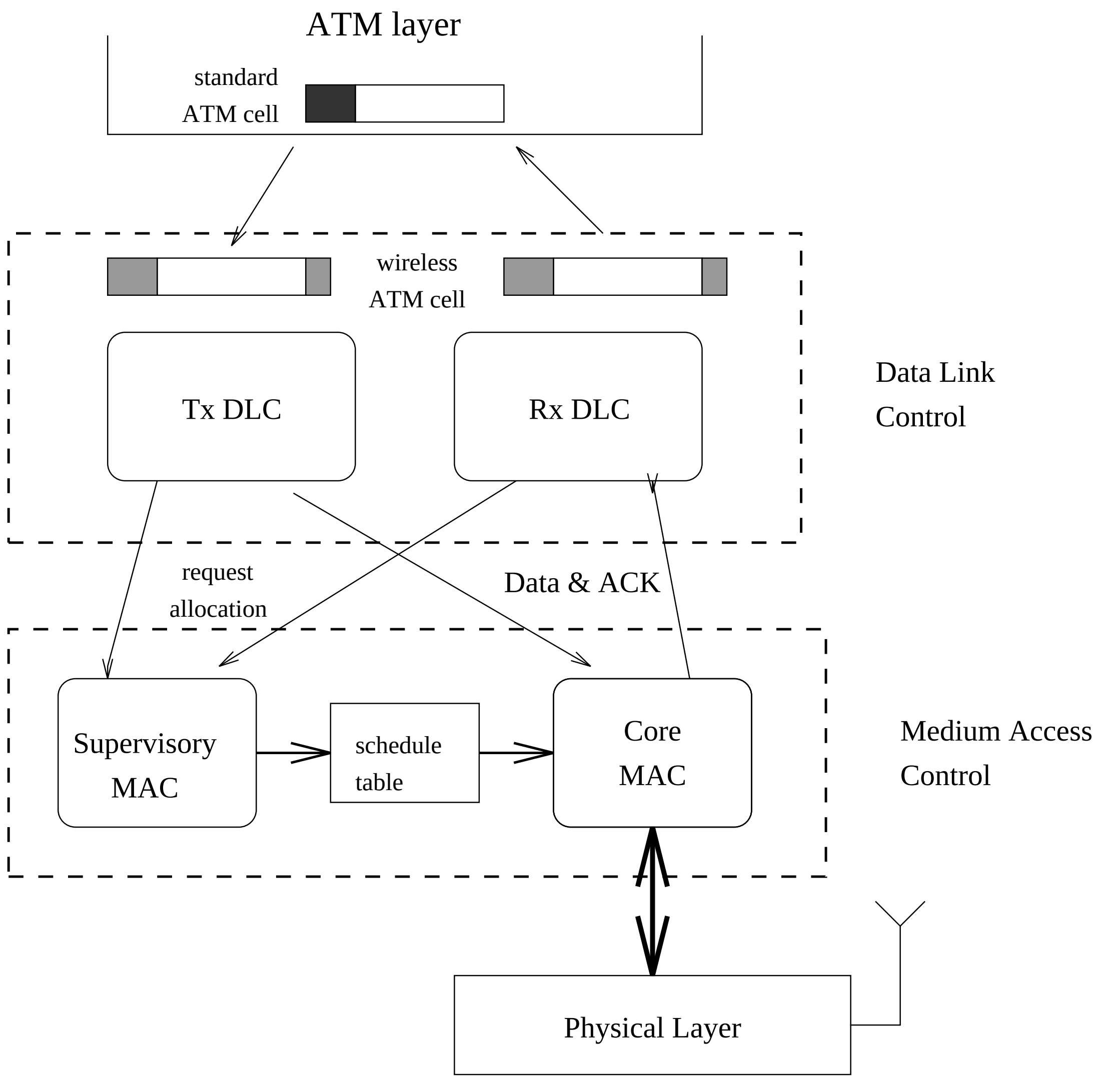


Figure 5.2: Architecture of the DLC and the MAC protocol layers

5.2.1 Cell Format

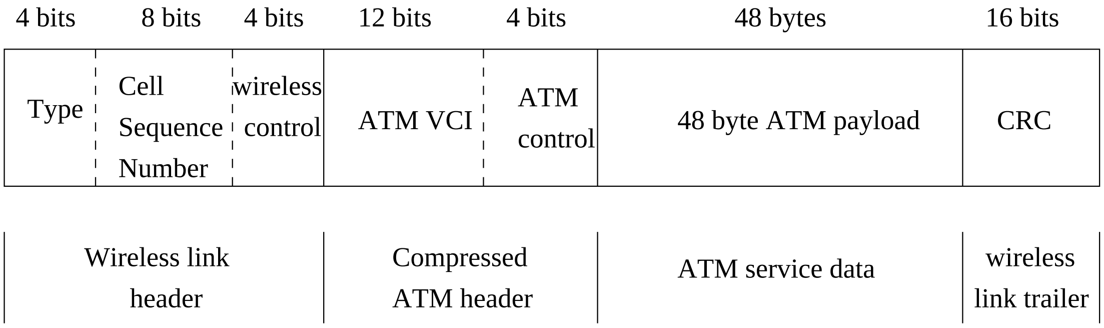
The cell format is harmonized with the standard ATM cell format [54]. Figure 5.3 shows typical wireless and ATM cell headers corresponding to the wireless ATM protocol stack shown in Figure 5.1. The wireless header incorporates an 8–bit cell sequence number to support DLC error recovery over moderately large bursts. This header also includes fields to enable other wireless network functions such as service type definition, handoff recovery, and cell segmentation. A 2–byte compressed ATM header is incorporated with 12 bits for VCI (virtual circuit identifier) and 4 bits for ATM control. The header error control (HEC) function is not used on the radio link in view of a 2–byte CRC error detection provided by the DLC. Note that optionally, the full 5–byte ATM header may also be retained if wireless channel efficiency is not too critical.

The format of a wireless ATM acknowledgment (ACK) message is also shown in figure 5.3. A maximum of 20 cells can be acknowledged with a single WATM ACK message. A bitmap of 20 bits in length is used to acknowledge cells offset from the cell sequence number (CSN) contained in the ACK packet. This type of group ACK syntax is used to reduce the number of short ACK packets requesting channel access from the MAC layer, thereby improving efficiency.

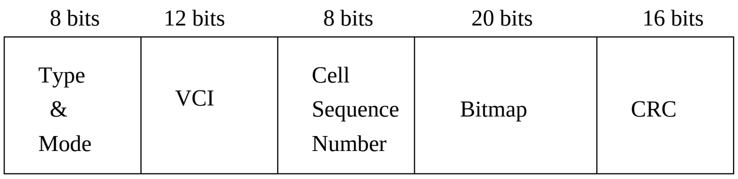
5.2.2 Frame Structure

Figure 5.4 shows a TDD/TDMA channel format that will be used to support the traffic classes outlined above. The channel frame contains downlink preamble, control, and data, followed by uplink control, ABR data, VBR data, and CBR data. Downlink transmissions are multiplexed in a single burst from the base station to the remote stations, while uplink transmissions use dynamic TDMA. The boundaries between the subframes are varied gradually with time in order to support changing traffic needs.

A brief description of the different subframes and their functions is given below. The downlink preamble provides frame synchronization as well as a training sequence for the purposes of equalization. The base station uses the frame header to transmit its ID and information about the positions and sizes of the different subframes in the



Wireless ATM cell format



Wireless ATM ACK packet format

Figure 5.3: Format of a Wireless ATM cell and a Wireless ATM acknowledgment (ACK) packet

current frame. The downlink control messages, transmitted in the Base–to–Remote (B–R) control region, include allocation information, B–R acknowledgment information, and other wireless network control messages (for example, messages for power control, handoffs, etc.). The B–R data region contains downlink data cells. The remote stations are informed about possible data cells to be received by them by via allocation messages transmitted in the B–R control region (except in the case of CBR calls where the positions are static, as described above). The Remote–to–Base (R–B) control region is used by the remote stations to transmit control messages to the base station. These control messages include slot allocation requests for R–B ABR bursts, wireless network control messages, etc. These messages are sent in a slotted ALOHA mode, where the remote stations randomly choose a slot to transmit their message and employ a backoff scheme to retransmit unsuccessful packets. The uplink data and R–B acknowledgment messages are transmitted in the R–B data region. The transmissions in this region are either allocated on a frame–by–frame basis (in the case of ABR transmissions, CBR/VBR retransmissions, and R–B acknowledgments) or pre-assigned in the case of allocation for CBR sources.

5.2.3 Medium Access Control (MAC)

Functionally, the MAC (Figure 5.2) can be described as consisting of two different components – a Supervisory MAC (S-MAC) and a Core MAC (C-MAC). The functions of these two components are as follows :

- **Supervisory MAC** : The S-MAC can be thought of as the "intelligence" of the MAC layer. It processes control information and builds a 'schedule table' for the C-MAC in each frame. At the base station, the S-MAC is responsible for channel access scheduling, both uplink and downlink, (except for the R–B control region) for all the VCs (ABR/CBR/VBR) in the system. The S-MAC receives requests for slot allocation from the VCs present in the system. Subject to the fulfillment of QoS requirements of each VC, the S-MAC arrives at a policy of scheduling data transmissions and acknowledgments (to enable error recovery by the DLC).

Call admission control (CAC) functions for the wireless link is also part of the base station S-MAC. At the remote station, the S-MAC processes the control information received in each frame in the B–R region and builds a schedule table for the C-MAC. It is also responsible for scheduling the transmission of the R–B control messages in the slotted ALOHA mode.

- **Core MAC** : The C-MAC can be thought of as the interface between the DLC and the physical layer. Based on a schedule table supplied by the S-MAC, it multiplexes/demultiplexes transmission and reception for the corresponding VCs. A schedule table entry, typically contains service type (transmission/reception), message type (data/control message), the VCI, the position and the duration of the service. The functions of the C-MAC at both the base station and the remote station is more or less identical. The functions of the C-MAC make it suitable for fast hardware implementation.

The S–MAC at the base station handles all the bandwidth allocation functions. This is the main functional difference from FRMA described in Chapter 3. It receives requests from the VCs at both the base station (for downlink transmission) and the mobile stations (for uplink transmission). The bandwidth allocation is class–based (CBR, VBR, or ABR). A brief description of possible bandwidth allocation strategies follows.

CBR VCs are assigned slots periodically according to their bit–rate. A group of $M \geq 1$ frames (e.g. $M = 6$) may constitute a MAC superframe, which is used to obtain a reasonable granularity for bit–rate assignments (e.g. $N \times 32$ Kbps). The positions of the assigned slots within a MAC superframe are maintained relatively static in order to facilitate operation of low complexity telephone terminals and also to reduce the control signaling load on the wireless link. Once the allocation for CBR calls are transmitted to the remote stations, they are assumed to remain in the same positions until they are explicitly changed by another allocation message by the base station. The positions of the slots for the CBR calls may need to be changed in order to achieve packing efficiency and lower the CBR call blocking probability. However, the time horizon over which such changes are effected is much longer than the duration of a MAC frame.

ABR virtual circuits (VCs), on the other hand, are handled on a burst–by–burst basis with dynamic reservation of ABR slots and unused CBR/VBR slots in each frame. There are advantages to be had, especially in the Remote–to–Base (R–B) ABR transmission, with contiguous allocation of slots for cells in a single ABR burst because of reduced overhead. A round–robin slot allocation policy may be used to prevent long bursts from overloading the channel and increasing the delays for the other VCs. ABR slot assignments are demand–based in contrast to the CBR slot assignments which are periodic. So allocation information consisting of the position and duration of the ABR slot assignment needs to be transmitted to the remote stations whenever these slot assignments are made.

VBR VCs are assigned slots with the aid of a usage parameter control (UPC) based statistical multiplexing algorithm. In order to improve channel utilization, this algorithm is augmented by an 'instantaneous' bit–rate estimation at the base station over an estimation interval whose size is of the order of a MAC superframe discussed above. A discussion of a novel scheme for bandwidth allocation for VBR sources such that each source achieves its QoS (cell loss, in this case) is discussed in chapter 6. Note that CBR and VBR calls may be blocked, while ABR virtual circuits are always accepted, subject to source rate flow controls.

5.2.4 Data Link Control

As described above, the Data Link Control (DLC) in wireless ATM is designed to provide an additional layer of error protection to ATM services over the radio link. Service class specific retransmission strategies are used to improve error recovery without sacrificing the wireless channel utilization. For ABR, DLC operation follows traditional SREJ ARQ procedures on a burst-by-burst basis, without time limits for completion. On the other hand, for CBR and VBR, each retransmission effort lasts till a time limit that is specified by the application at call set-up time. For CBR and VBR, the channel allocation is periodic; additional ABR allocations are made at the MAC layer to support retransmissions. Since the retransmissions use additional bandwidth, the system can attain a better quality of service without affecting the normal operation of CBR and VBR traffic.

Each DLC layer has two independent components, one to handle the DLC functions of all the cells that are to be transmitted out of the station (remote or base), and the other to handle the received cells. They maintain status information for all of the cells in the current active burst. In the transmission mode (Tx), this is necessary to determine the cells that need to be transmitted/retransmitted, and those that have been transmitted successfully[2] and hence can be deleted and the buffer released. The DLC maintains a Tx table, where the status information of the cells in the buffer are maintained. Cells that are not positively acknowledged by the receiving station are retransmitted at the next available opportunity. In the reception mode (Rx), the DLC maintains status information in order to send proper acknowledgments (ACK) to the transmitting station which can then retransmit all the lost cells. The Rx DLC maintains a Rx table to achieve this. Cells received successfully are sent to the ATM layer in a manner appropriate to the traffic class.

Finally, the functions of the DLC layer differ depending on whether it is at the base station or at the remote station. This stems from the fact that all the base station controls all the slot allocation (including retransmission requests). So the DLC layers at the base station need to maintain extra information to submit requests for retransmissions.

5.3 System Performance

We built a detailed software emulator to test the functioning of the wireless ATM protocols described. We implemented the MAC layer and the DLC layers for ABR and CBR traffic classes. With the aid of the software emulator, we evaluated the performance of the MAC and the DLC layers and present these results in this section.

We chose the basic unit of MAC allocation to be a 8–byte slot. A 16–byte preamble was required for synchronization and equalization at the start of every burst. Since the

[2]A cell from an ABR VC is considered to have been successfully transmitted if it has been received without any errors at the receiving station and has been acknowledged.

base station transmits all its information in one burst, we have a 16–byte B–R preamble at the start of every frame. Every isolated burst of transmission from a remote station (data/control) requires a preamble of 2 slots. Data cells required 7 slots each (48–byte payload, 4–byte WATM header, and 2–byte CRC) and control packets 1 slot each (8 bytes containing a 2–byte header, 4–byte control payload, and 2–byte CRC).

The system parameters used in the performance evaluation are summarized in Table 5.1. As can be seen, 20% of the bandwidth is allocated to the two control regions. So the maximum achievable data throughput (under ideal conditions, without overhead) is limited to 80%.

Channel bit rate	8 Mbps
Frame duration	2 ms
B–R control	8%
R–B control	12%
Slot size	8 bytes
Slots per frame	250
Preamble size	2 slots
Frame header	2 slots
Data cell	7 slots
Control packet	1 slot

Table 5.1: System parameters used in performance evaluation.

5.3.1 ABR performance

The first phase of the simulations included ABR sources only. The system consisted of a base station and 15 mobile terminals. Each of the terminals had 8 ABR VCs initialized and hence the system had 120 bidirectional ABR VCs initialized. The ABR burst size, L (cells), was geometrically distributed with mean value $E[L]$, and when the transmission of a burst was completed, an idle period (before next burst arrival), distributed exponentially with a mean of τ, followed. The offered load is varied by using different τ values.

In Figure 5.6, the mean ABR burst delay[3] is plotted as a function of the mean

[3] The ABR burst delay is defined as the time interval (measured in frames) between the start of

burst length. We assumed the channel to be error-free, *i.e.* reflecting the effects of MAC operation alone. For lower values of the average burst length, we decreased the mean idle time between bursts proportionately in order to keep the offered load[4] on the system comparable for the three values of the burst length. It can be seen that the maximum throughput[5]depends on $E[L]$. When the mean burst size is small, e.g. $E[L] = 1$ or 2, the system needs more bandwidth for B–R control channel and random access channel for each data cell, and the utilization of the data channel is thus limited by the congested control channel. When $E[L] = 5$ or 10, the control channel is less congested, and can achieve higher throughput. When the mean burst length exceeds 10 cells, the delay–throughput curves are not distinguishable from each other. The maximum achievable throughput, when the mean burst length is 10 cells or higher, is close to the ideal throughput of 80%. Also, on the average, the Base–to–Remote bursts experienced slightly lower delays than the Remote–to–Base bursts since the latter have a random access contention delay component.

Figure 5.7 plots the maximum throughput as functions of cell-error rate P_{cell} given $E[L] = 1, 2, 5, 10$ respectively. The bit error rate (BER) was assumed to be constant from frame to frame. The maximum throughput decreases as P_{cell} increases because a higher cell error probability implies more lost cells, which have to be recovered by retransmissions. The results show that, with very small BER, the maximum throughput reaches 0.74, the 0.26 overhead is from frame header, preambles and control traffic. As P_{cell} increases, the retransmission traffic increases also, and the actual throughput decreases. We derived a channel throughput upper bound, shown as follows.

Let $S^*_{\max}$ denote the maximum throughput that can be achieved given $P_{cell} = 0$ and no congestion in the control channels. Under this condition, we assume that the data channel is almost 100% utilized (except some remaining idle slots that are too short to carry a data cell). Given the system parameters in Table 5.1, we can derive $S^*_{\max} = 0.78$.

---

the burst and the instant when the last cell of the burst has been successfully acknowledged by the receiving station.

[4]The average number of number of cells arriving into the system, normalized by the channel rate.

[5]The average number of information bits (wireless ATM cell, including header) per frame, normalized by the total number of bits per frame.

With $P_{cell} > 0$, the maximum achievable throughput $S(P_{cell})$ as a function of P_{cell} is given by

$$S(P_{cell}) = (1 - P_{cell} - P_{ctl} - P_{hdr}) S^*_{\max}, \tag{5.1}$$

where P_{ctl} and P_{hdr} are the error rate of a B–R control packet and the error rate of a frame header, respectively. Given a value of BER, we can derive the values of P_{ctl} and P_{hdr}. The formula gives the upper bound of ABR throughput, which can be reached only when the system has one heavy user (the "Ideal" curve) with a burst of infinite size. Figure 5.7 shows that, with larger $E[L]$ and small P_{cell}, the maximum achievable ABR throughput is close to the upper bound. When P_{cell} increases, the number of retransmissions increases, and the B-R control channel may be congested frequently, hence, the achievable throughput drops.

A fading model [23] for the channel was used to study the effects of log–normal fading on the performance of the system. The signal and interference have a shadow fading with a standard deviation $\sigma = 3.3\ dB$. We assumed that the interference was caused by the six nearest cells that were using the same frequency. The total interference was modeled as the sum of the interference from these six cells, along with a log–normal fading component. We used a path loss coefficient of $\alpha = 3$ inside the cell and $\alpha = 3.5$ for interference from outside the cell. A hexagonal cell structure with a reuse factor of 3 was assumed. A first–order autoregressive model was used to model the fading signal strength. To determine the current BER for a VC, we first calculate the signal-to-interference ratio (SIR) using the fading model, then approximate the interference as the white noise with the same power to find the symbol error rate for $\pi/4$ QPSK modulation. In the physical layer, (63,51) BCH code is used, which can correct up to 2 symbol errors in every 63 symbols, and the BER can be calculated by taking the error correction capability into account. Figure 5.8 shows the ABR message delays vs. the autocorrelation factor ρ for $E[L] = 5, 10, 20$, respectively. Here, a large ρ represents a model with deep fading (once the signal enters a fade, it has high chance of staying in the fade during next frame), and the ABR services have longer waiting times.

5.3.2 CBR Performance

In the second phase of performance evaluation, CBR sources were included in the system in addition to the ABR sources. In CBR services, the quality–of–service measure of interest was the released cell-loss rate after DLC processing. ABR and CBR traffic are mixed without any MAC layer service priority between ABR and CBR retransmissions, ABR load is maintained at 15% – 20%. A total of 15 remote stations were present in the cell, with 5 of them engaged in a CBR call each. Each remote station had 6 ABR VCs, thus bringing the total number of ABR VCs to 90. All the VCs were bi–directional and the traffic symmetric. The CBR bit–rates shown are the net data rate (excluding the cell header and any preamble).

Figure 5.9 depicts the cell-loss rates as functions of cell-error rates under different CBR loads, given 40 ms as the DLC buffering time. Note that under heavy load condition, the cell-loss rate is higher because the CBR retransmissions were in the ABR mode and, hence, have to compete with the ABR sources sharing the channel. The results shown in Figure 5.9 assumed no priority for the CBR retransmissions over the ABR sources. When CBR retransmissions were given priority over the ABR transmissions/retransmissions[6], we found that the error–recovery of CBR cells was total for BER values of 10^{-4} and lower, even at high traffic loads.

In Figure 5.10, the CBR DLC cell buffering delay was varied in order to study its effect on the performance. The BER was a constant, corresponding to a cell error rate of 0.0115. When the CBR load was high, a short DLC buffering time[7], T_{DLC} for the CBR cell resulted in almost no recovery. A higher bandwidth of the CBR calls resulted in a lower bandwidth for the retransmission. Hence, the ability of the system to recover lost cells decreased rapidly with a short lifetime of the CBR cell. But at lighter loads, the cell recovery was good even with a short lifetime of the cell. Given a longer lifetime (e.g. > 20 ms) and at lighter loads, all the cells with error were recovered by retransmissions.

---

[6]This was achieved by having no ABR sources in the system.

[7]The amount of time before the information content in the cell becomes insignificant. This could also be considered as the lifetime of the cell.

In Figure 5.11, we plot the CBR cell loss probability as a function of the ABR throughput for different CBR call bandwidth requirements (assumed symmetrical in B–R and R–B directions). There were 5 CBR calls each requiring the bandwidth indicated in each of the two directions. At any particular CBR load, the higher the ABR load on the system, the higher the CBR cell loss probability. The CBR re–transmissions did not have priority for slot allocation over the ABR sources. The higher the CBR call bandwidth requirement, the lower the bandwidth available for ABR sources and CBR re–transmissions. Hence, with an increase in the ABR load, the CBR cell loss increases.

As noted earlier, giving the CBR retransmissions some priority over ABR traffic significantly improves the cell recovery of the CBR sources at the cost of QoS for the ABR sources. When the CBR cell buffering time is short (for example 10 ms), a priority scheme for CBR retransmissions over ABR may be required to attain the quality of CBR services.

5.4 Conclusions

The objective of the proposed wireless ATM system is to provide integrated multiservices for ABR and CBR/VBR while achieving a specified quality of service. Due to the inherent high error rate of the wireless channel, a DLC protocol is essential for approaching ATM quality–of–service comparable to that achievable in conventional ATM networks. DLC protocols for ABR and CBR service classes have been described in this chapter. We have also presented performance evaluation results from a software emulation of the wireless ATM network. They show that, for the traffic and channel models under consideration, useful throughput of about 60 to 75% of the channel rate can be achieved while maintaining reasonable QoS in a system with both ABR and CBR sources.

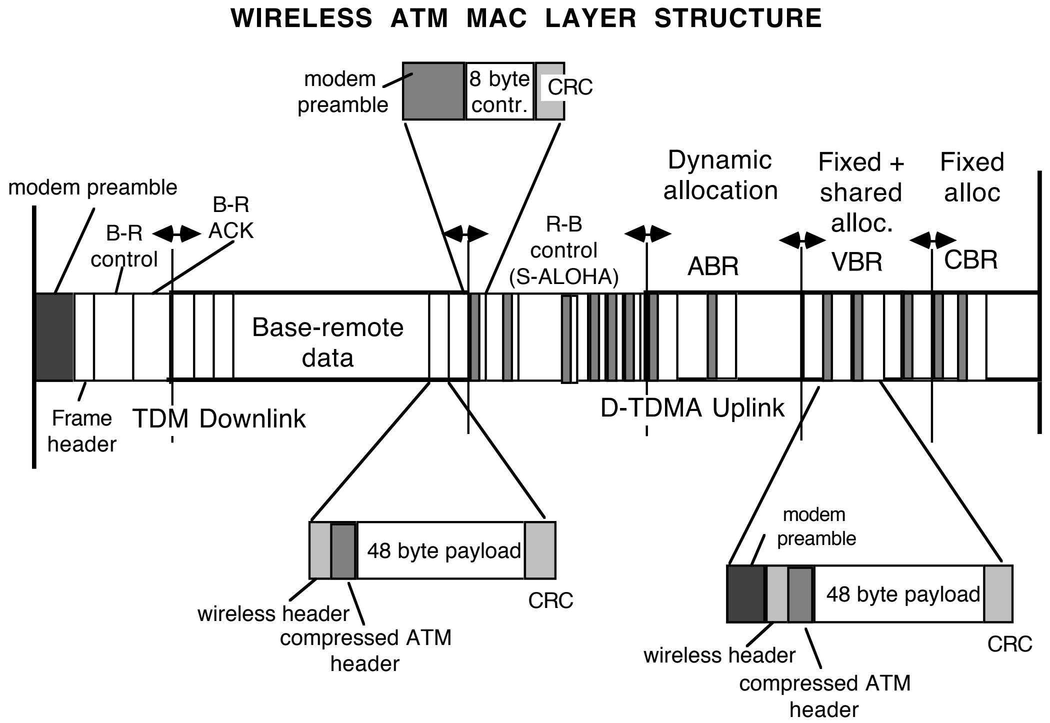


Figure 5.4: Wireless ATM frame structure

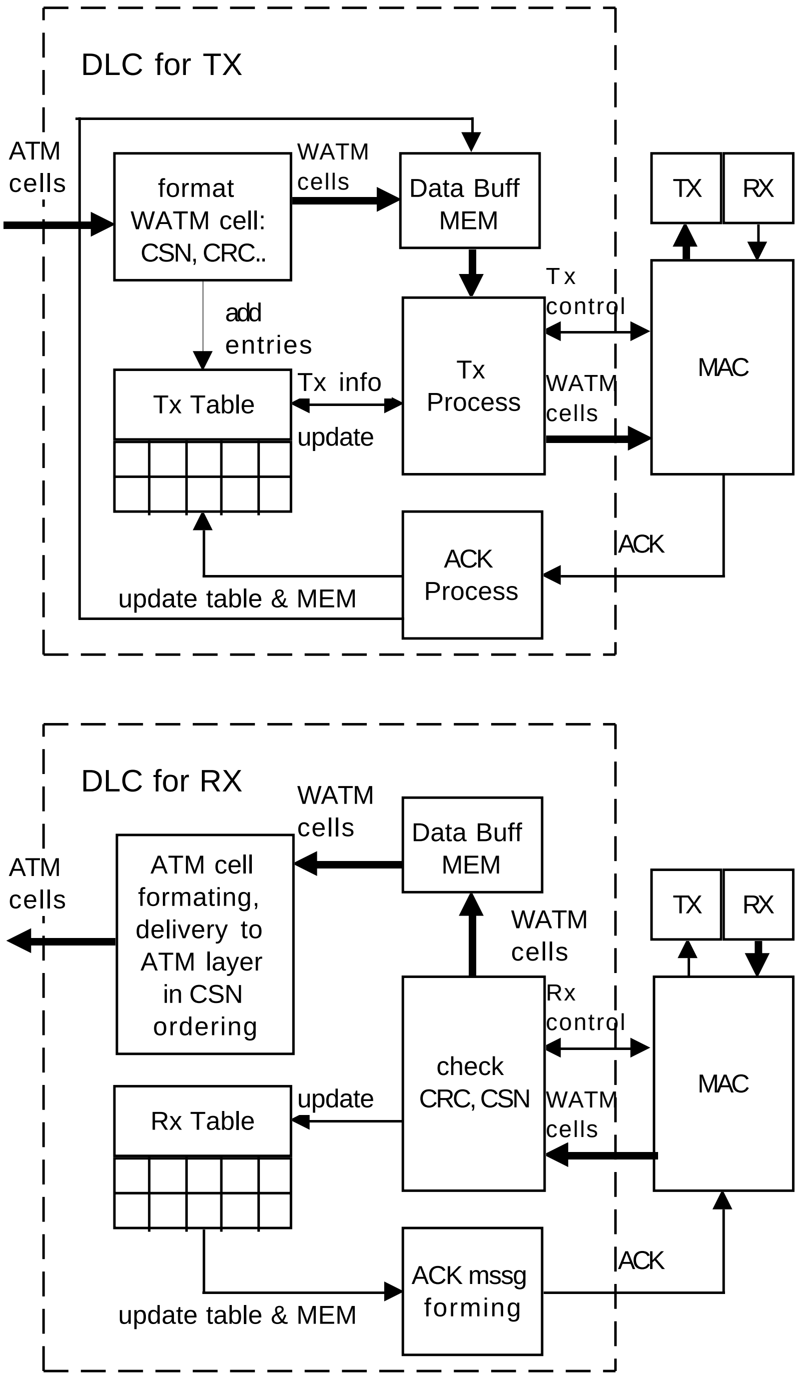


Figure 5.5: Architecture of the DLC layers

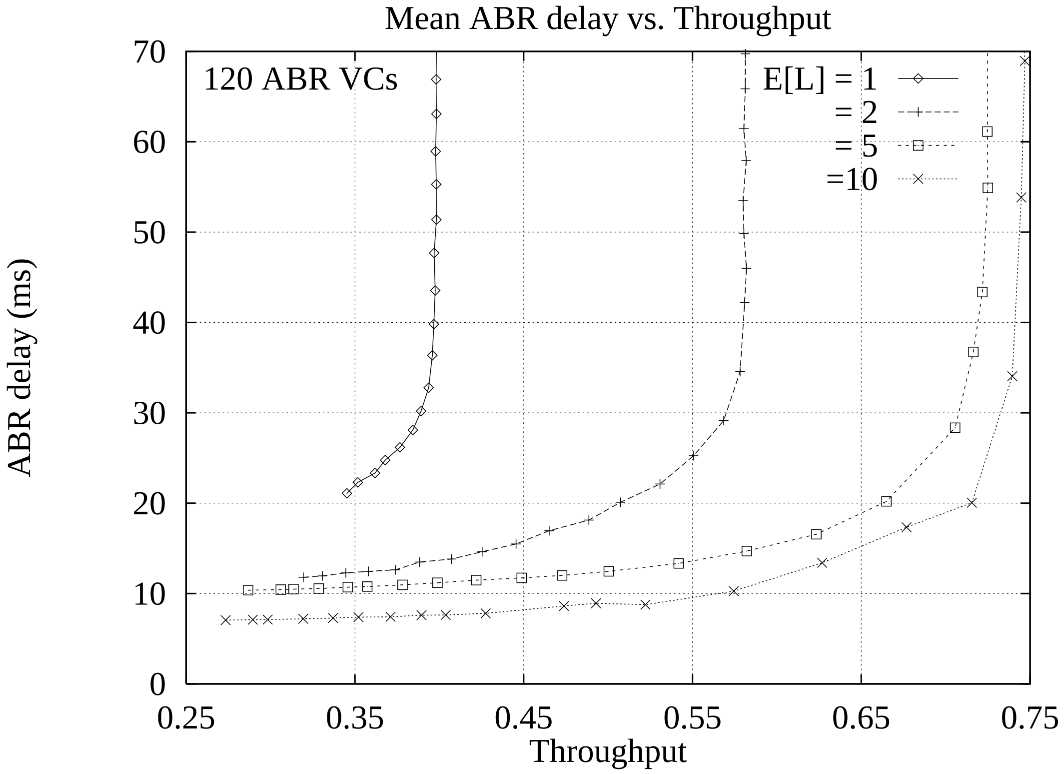


Figure 5.6: ABR burst delay as a function of the total throughput for different values of the mean burst length. The channel was error–free.

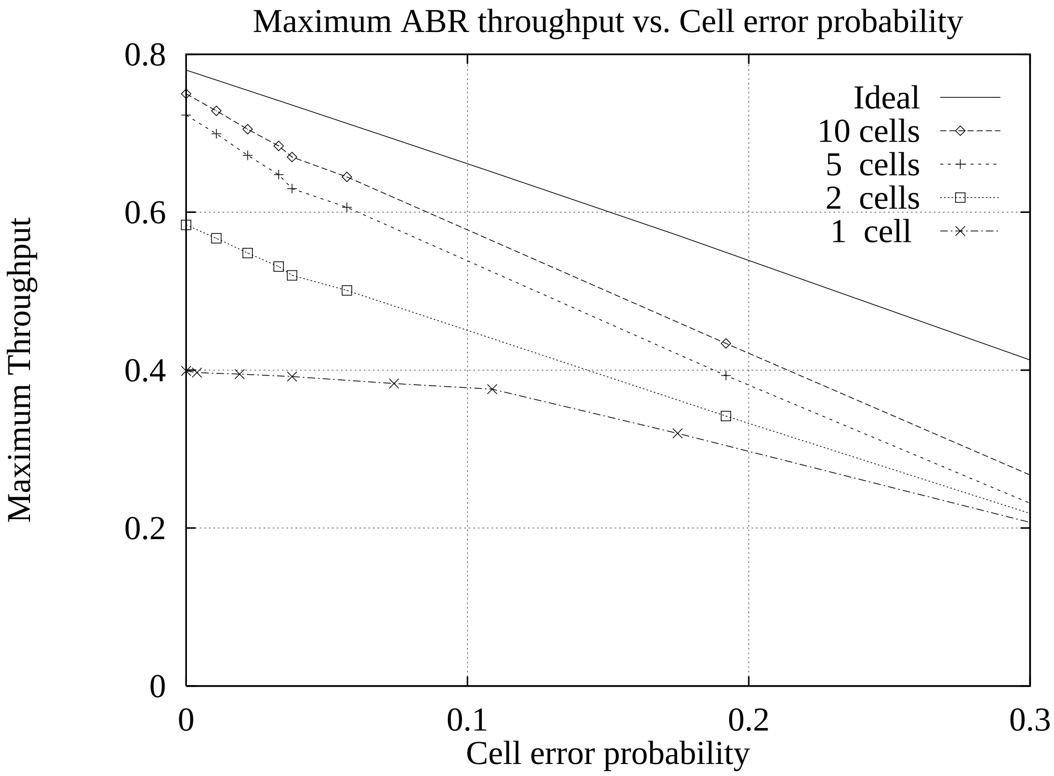


Figure 5.7: Maximum ABR throughput as a function of the cell error rate for different values of ABR burst length.

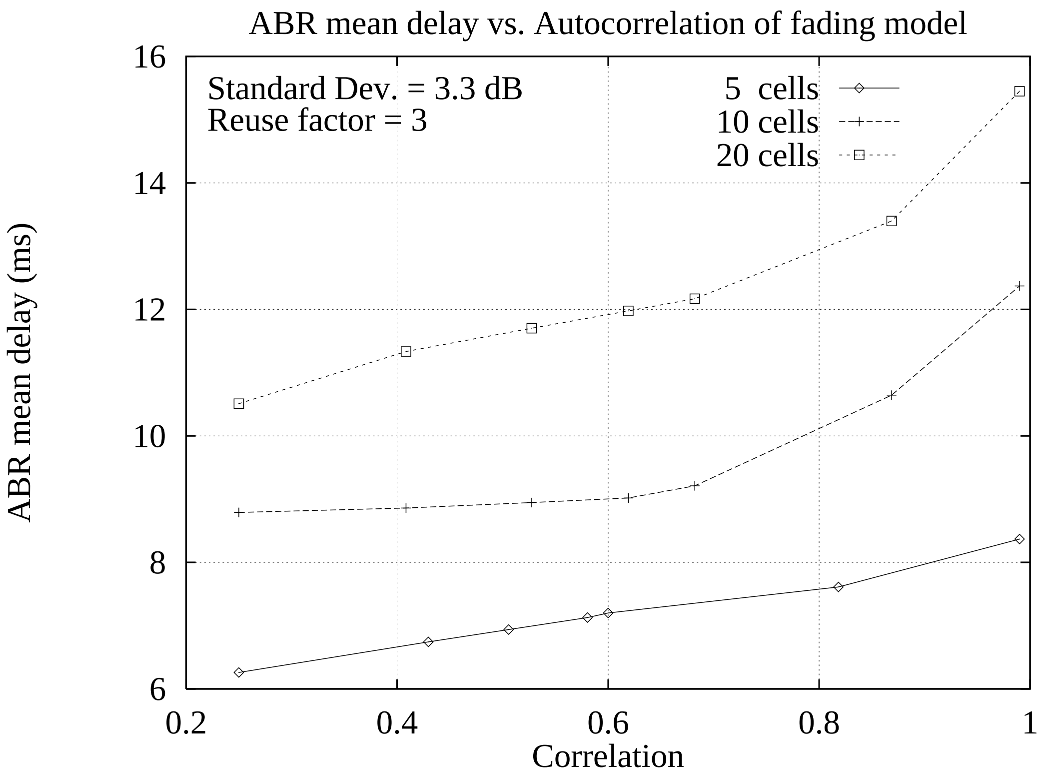


Figure 5.8: The mean ABR burst delay as a function of the autocorrelation value used in the autoregressive function used to model fading.

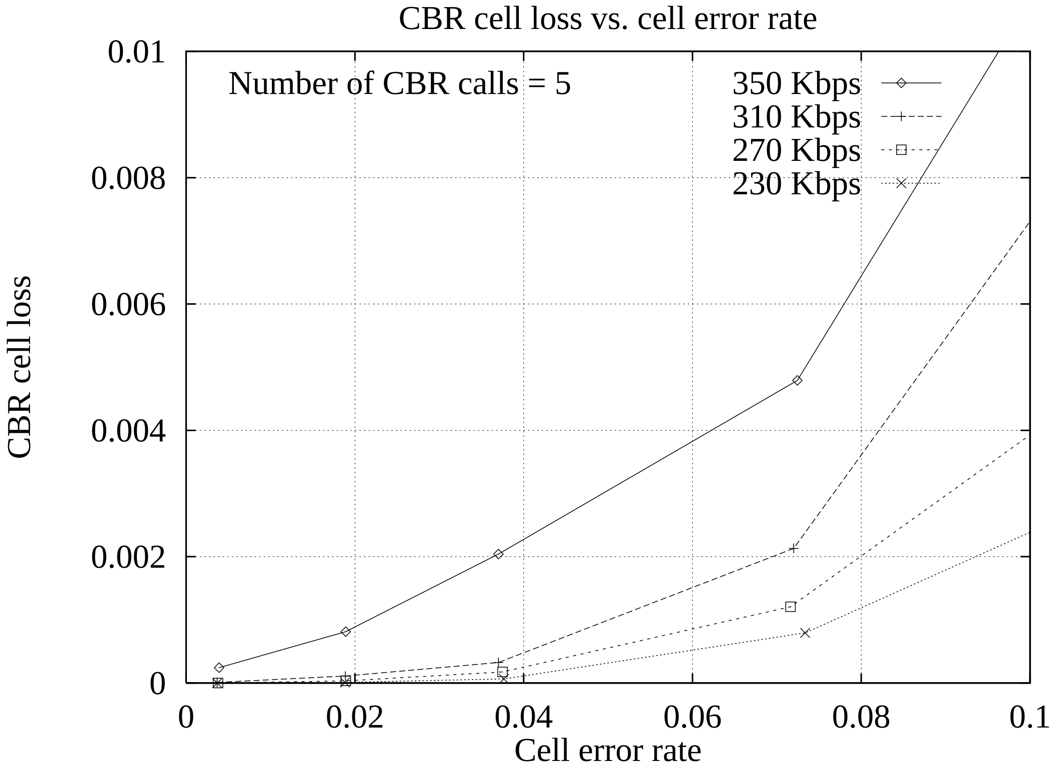


Figure 5.9: CBR cell loss probability as a function of the CBR cell error probability for different values of bit–rates of each CBR call. 5 CBR VCs, $T_{DLC} = 40$ ms.

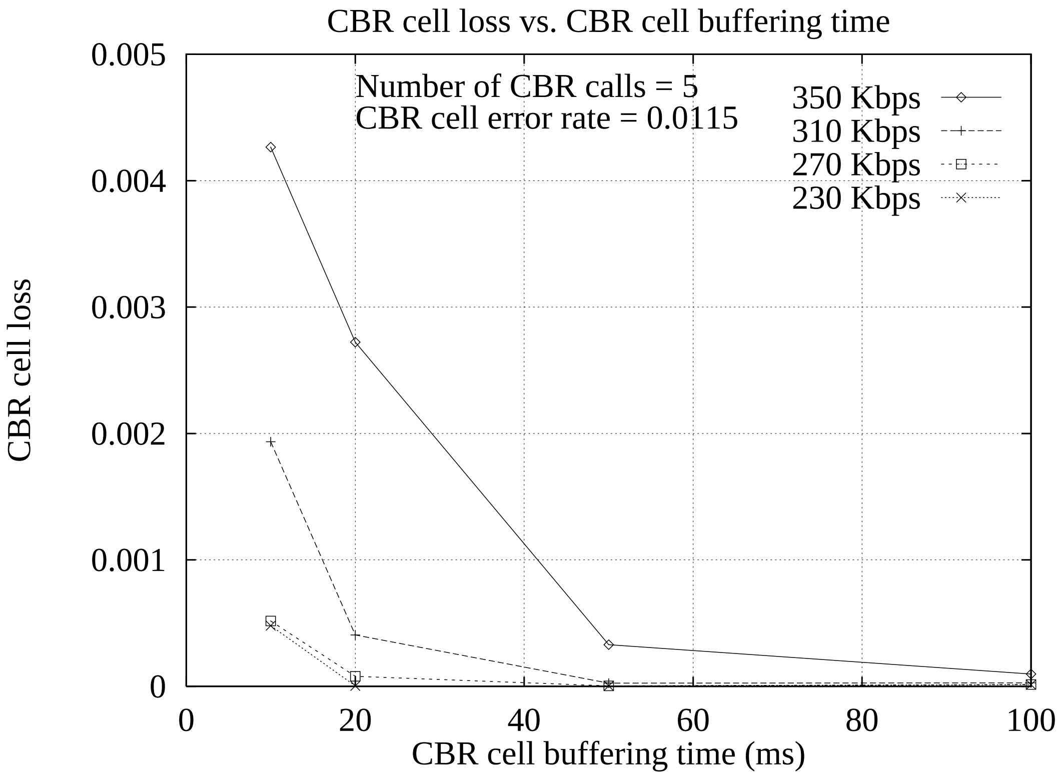


Figure 5.10: CBR cell loss as a function of the T_{DLC}. 5 CBR VCs, ABR load between 0.15 to 0.20.

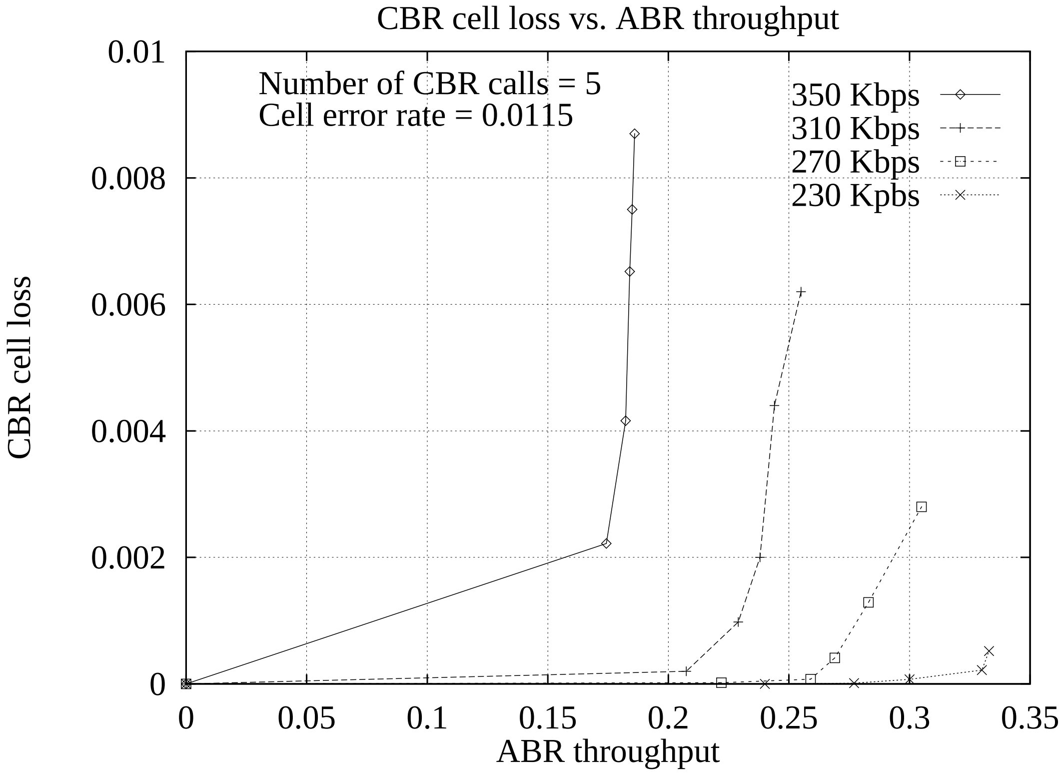


Figure 5.11: CBR cell loss as a function of the ABR load for different CBR call bandwidth requirements. 5 CBR VCs, $p_{cell} = 0.0115$.

Chapter 6

Centralized Medium Access Control

In PRMA, the base station is not expected to perform any active bandwidth allocation function at the MAC layer. It is a passive listener, that provides feedback to the mobile terminals, on the downlink, based on the information received and decoded on the uplink. In FRMA, the base station exercises a limited amount of control on how the bandwidth is split among the voice, voice contention, and the data regions. But these decisions are based on the voice contention process and the base station consults a look–up table to determine the bandwidth allocation for the current frame.

In the wireless ATM system, described in chapter 5, the base station completely controls the bandwidth allocation at the MAC layer. The base station is assumed to have perfect information on the bandwidth requirements of each of the information sources connected to the system. The base station takes into consideration the QoS requirements of each of the sources and partitions the available bandwidth appropriately. Going back to the classification of multiaccess protocols, we could say that PRMA is a hybrid of a random access technique and a fixed assignment technique. The multiaccess part of the wireless ATM system falls into the category of demand assignment techniques with the assignment by a central authority (the base station, in this case).

In this chapter, we study centralized bandwidth allocation schemes at the MAC layer, where the base station plays an active role in the manner in which bandwidth is allocated from frame to frame. This study was motivated by the problem of allocating bandwidth to variable bit–rate (VBR) sources in the wireless ATM system. The VBR sources generate information, whose bit–rate is time–varying, in contrast to the constant bit–rate (CBR) sources which generate information at a constant rate. An often quoted example of a VBR source is compressed video, where the bit–rate varies from one video frame to another.

The information being generated by the sources in the system is assumed to be time–critical in nature. That is, if the information is delayed beyond a certain time–limit, then it loses its value and is dropped. So there is no queuing at the source beyond the specified time–limit. This is similar to the assumptions in FRMA where the speech packets have a lifetime of about the duration of a frame. So, packet[1] loss probability will be the QoS parameter of interest.

Even though, this problem was motivated by the bandwidth allocation for VBR in the wireless ATM system, the results from this chapter could be used to determine resource allocation in any system where, on a periodic basis, a central arbitrator uses information from a given number of users requesting variable amounts of the resource and to partition the available resource. The sum of all the requests could exceed the number of units of the available resource and some users may be allocated fewer units of the resource than they had requested. There would be a penalty for allocating fewer units of the resource than a user had requested and an objective, similar to the one described in this chapter, has to be met.

Parekh and Gallager [46, 47] present an approach to allocating network resources to the users of an integrated services network. Their approach, called Generalized Process Sharing (GPS), is a generalization of uniform processor sharing [29]. It is characterized by positive real numbers, $\phi_1, \phi_2, \ldots, \phi_K$, ($K$ is the number of sources in the system), and if $S_i(t_1, t_2)$ is the amount of service that source i receives in the interval $(t_1, t_2]$, then

$$\frac{S_i(t_1,t_2)}{S_j(t_1,t_2)} \geq \frac{\phi_i}{\phi_j}, j = 1, 2, \ldots, K. \tag{6.1}$$

If r is the rate of service of the server, then source i is guaranteed a rate of

$$g_i = \frac{\phi_i}{\sum_{j=1}^{K} \phi_j} r. \tag{6.2}$$

The assumption here is that sources can be backlogged and the information is delayed

[1]The terms "packet" and "cell" will be used interchangeably to denote the unit of information. Also, the duration of a time slot is the transmission time of one packet/cell.

when resources are not available in sufficient quantity and there is no loss of information. Worst–case packet delay and worst–case source backlogs are derived for GPS when combined with Leaky Bucket admission control [60]. They also present a packetized version of GPS, called PGPS, that can be implemented on packet communication systems. The PGPS is very similar to another scheme called Weighted Fair Queuing, proposed in [11].

GPS and PGPS were developed for wireline ATM networks and represent efforts to control access and avoid cell loss at a downstream multiplexer. GPS is an open loop control system since propagation delays prevent the use of feedback control. In our system, propagation delays are small. Also, there is an explicit MAC layer that controls access and hence facilitates closed loop control. The other difference between the two systems is that our system suffers information loss rather than message delay as in GPS/PGPS.

The rest of the chapter is organized as follows. In Section 6.1, we describe the system model used in the analysis of the centralized bandwidth allocation schemes described in this chapter. In section 6.3, we describe two criteria for dropping packets, in frames where packet loss is inevitable. In section 6.4, we show that a greedy algorithm solves the optimization problems associated with the above–mentioned criteria for dropping packets. We conjecture the convergence of the normalized QoS if either of the two criteria for dropping packets is implemented and describe the conditions under which this could occur in section 6.5.

6.1 System Model

The model for the system that will be used in the study of the centralized MAC is as follows

- The base station allocates bandwidth (time slots, in this case) once every frame. The amount of available bandwidth is a constant, *i.e.,* the number of available slots per frame is a constant.

- The information sources are assumed to communicate their bandwidth requirements for a particular frame before the start of that frame. Typically, the information is transmitted in the frame preceding the current one. The information sources could be either at the base station or at the mobile terminal. Since the base station is the entity allocating bandwidth, the sources at the base station can communicate their requirements with an internal call to the bandwidth allocating entity (the S–MAC in the wireless ATM system). However, the information sources at the mobile terminals have to do so using the radio link. The use of a control channel is assumed to aid this. No assumptions about the nature of this control channel are made at this time. A small portion of the total bandwidth could be allocated for this purpose, whereby the mobile terminals transmit their requirements. On the other hand, a few bits in the header of each packet/cell could be utilized for the same purpose too.

- The bandwidth requirements are in the form of an integral number of time slots required in the next frame.

- The information generation process at each source is independent of those at the other sources.

- The sum of the peak bit–rates at which information is generated by the sources is assumed to be greater than the available capacity. That is, the number of packets offered to the system in any particular frame by all of the information sources is sometimes greater than the number of available slots. Otherwise, the bandwidth allocation problem is trivial since all the packets offered will be accepted for transmission and there would be no packet loss.

- When the total number of packets/cells offered to the base station by all the sources in a frame is less than or equal to the number of available slots, then all the packets are accepted for transmission.

- However, if the total number of offered packets exceeds M, the number of available slots, then the base station allocates bandwidth in such a way that a total of M

packets are accepted for transmission. All packets/cells that are not accepted for transmission in a current frame are assumed lost. Hence, the base station has to decide which sources will have packets dropped.

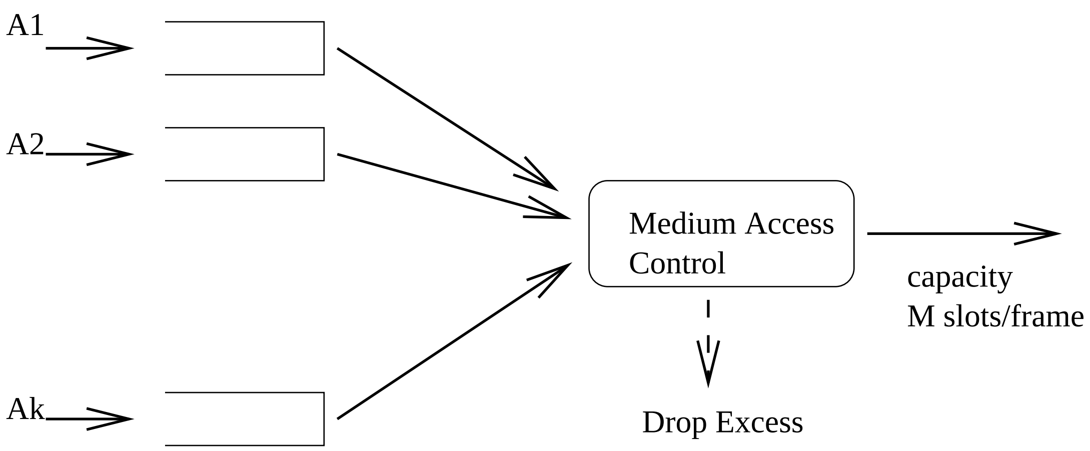


Figure 6.1: Centralized MAC

Since we have assumed that the number of packets offered to the system in a frame will sometimes exceed the total number of available slots in a frame and that if all the offered packets cannot be carried in a frame, then those that are not accepted for transmission are lost, a certain amount of packet loss is inevitable. Lost packets correspond to lost information, and this affects the quality of the connection as perceived by the user. The user is aware of this possibility and announces an acceptable packet loss value at the time of setting up the connection. This value, termed the target QoS, signifies the maximum packet loss that is considered acceptable by the user. If the packet loss experienced by this connection exceeds this target QoS, then the service is unacceptable. The target QoS values specified by each user need not be the same.

Given that some packet loss is inevitable, and that the users have specified target QoS values, we address the problem of distributing the inevitable packet loss among the sources in the system subject to satisfying their target QoS constraints. We propose two criteria – the *minmax criterion* and the *maxmin criterion* – that the base station can use to partition the available bandwidth among the sources in the system. The criteria are aimed at, respectively, minimizing the worst normalized[2] packet loss faced by any source and maximizing the lowest normalized packet loss suffered by any source. The associated optimization problems can be formulated as mixed integer programming problems. We show that a greedy algorithm solves these optimization problems and, hence, implementation of these criteria is quite simple. We conjecture that, under certain conditions, these two criteria lead to a *even* partitioning of the available bandwidth among the sources in the system. Assuming that these conjectures are true, we can derive an expression to determine whether it is feasible to meet the target QoS values specified by the sources in the system with a given amount of bandwidth. This expression can be used in connection admission control (CAC) to decide whether a new connection could be accommodated without violating the feasibility of the QoS targets of the system already in the system and the one currently requesting a new connection.

6.2 Notation

In this study, we consider traffic in both directions, unlike the FRMA systems studied earlier where only uplink transmission was considered. Thus, the information sources in this study may be located either at the mobile terminal (and transmitting information to the base station, on the uplink) or at the base station (and transmitting information to a specific mobile terminal, on the downlink).

Let K denote the number of sources admitted into the system. We will use $[K]$ to denote the set $\{1, 2, \ldots, K\}$. M is the number of slots per frame that is available for transmitting packets from these K sources. For $i \in [K]$, let $w_i(t), t = 1, 2, \ldots$ denote the number of packets offered to the system by source i in frame t. We assume that,

$$\lim_{t \to \infty} \frac{1}{t} \sum_{y=1}^{t} w_i(y) = \lambda_i. \tag{6.3}$$

- Let $\delta_i(t)$ be the number of packets from source i that are dropped in frame t.

- Let $\Delta(t)$ be the total number of packets that are dropped in frame t, *i.e.*

$$\Delta(t) = \sum_{i=1}^{K} \delta_i(t). \tag{6.4}$$

[2]See Section 6.2 for a definition of this term.

Also,

$$\Delta(t) = \left(-M + \sum_{i=1}^{K} w_i(t)\right)^+, \tag{6.5}$$

where $(x)^+ = \max(x, 0)$. If the total number of packets from the K sources can be carried in the current frame, then none of the packets is dropped. Otherwise, the number of packets exceeding M are dropped.

- Let $n_i(t)$ be the total number of packets that source i has generated from the start of the connection (assumed to be at $t = 1$) through the end of frame t, *i.e.*

  $$n_i(t) = \sum_{y=1}^{t} w_i(y). \tag{6.6}$$

  From equation (6.3),

  $$\lim_{t\to\infty} \frac{n_i(t)}{t} = \lambda_i, \qquad i \in [K]. \tag{6.7}$$

- Let $d_i(t)$ be the total number of packets from source i that have been dropped from the start of the connection ($t = 1$) through the end of frame t, *i.e.*

  $$d_i(t) = \sum_{y=1}^{t} \delta_i(y). \tag{6.8}$$

- We will use τ_i to denote the target quality–of–service (QoS) that is desired by source i. We assume that the QoS is specified as the maximum number of packets that the source can drop per unit time while maintaining acceptable quality. The actual packet loss ratio that is acceptable to the source can be computed as τ_i/λ_i.

- Let $q_i(t)$ represent the QoS achieved by source i at the end of frame t, *i.e.*

  $$q_i(t) = \frac{d_i(t)}{t} \tag{6.9}$$

The normalized packet loss experienced by any source i is defined as the ratio of the current packet loss to the target QoS, $\frac{q_i(t)}{\tau_i}$.

6.3 Criteria for dropping packets

In this section, we describe the two criteria that the base station can use to determine which packets to drop (when all of the offered packets cannot be accepted for transmission in a frame). Then, we formulate the associated optimization problems as mixed–integer programming problems.

If the total number of offered packets in frame t is at most M, *i.e.*,

$$\sum_{i=1}^{K} w_i(t) \leq M, \tag{6.10}$$

then

$$\delta_i(t) = 0, \qquad i \in [K]. \tag{6.11}$$

Otherwise, if, in frame t,

$$\sum_{i=1}^{K} w_i(t) > M, \tag{6.12}$$

then, exactly M of the offered packets will be accepted for transmission. The excess, represented by $\Delta(t)$ (equation (6.5)) is the number of packets that will be dropped/lost in frame t. One of the two following criteria can be used to determine which of the offered packets should be dropped.

6.3.1 Minmax Criterion

The minmax criterion attempts to choose packets to drop in such a manner as to minimize the worst normalized packet loss faced by any source at the end of the frame. The decision variables, in this problem, are $\delta_1(t), \ldots, \delta_K(t)$. The minmax criterion can be represented as

$$\begin{array}{lll} \min_{\delta_1(t),\ldots,\delta_K(t)} & \max_{i\in[K]} & \frac{d_i(t-1)+\delta_i(t)}{\tau_i t} \\ \text{subject to} & & 0 \leq \delta_i(t) \leq w_i(t), \quad i \in [K] \\ & & \sum_{i=1}^{K} \delta_i(t) = \Delta(t) \\ & & \delta_i(t) \text{integer.} \end{array} \tag{6.13}$$

$d_i(t-1)$ is the total number of packets that source i has dropped through the end of frame $t-1$. Hence, $(d_i(t-1) + \delta_i(t))$ is the number of packets that source i will have dropped through the end of frame t. Dividing by $t\tau_i$ gives us the normalized packet loss experienced by source i at the end of frame t. The number of packets dropped by source i cannot exceed the number of packets offered by source i, which is $w_i(t)$. The total number of packets dropped should add up to $\Delta(t)$ and we require that $\delta_i(t)$ be integer for all i.

This problem can be re–formulated as the following mixed–integer programming problem:

$$\begin{array}{lll} \min & \beta & \\ \text{subject to} & \frac{d_i(t-1)+\delta_i(t)}{\tau_i t} - \beta \leq 0, & i \in [K] \\ & \sum_{i=1}^{K} \delta_i(t) = \Delta(t) & \\ & 0 \leq \delta_i(t) \leq w_i(t), & i \in [K] \\ & \delta_i(t) & \text{integer.} \end{array} \tag{6.14}$$

β represents the worst normalized QoS faced by any source. Since this is typically not an integer, and $\delta_i(t), i \in [K]$ are integers, this problem is a mixed–integer programming problem.

6.3.2 Maxmin criterion

The maxmin criterion for dropping packets is aimed at maximizing the best normalized QoS (or the lowest normalized QoS) faced by any source at the end of every frame. The constraint set is the same as for the minmax criterion. The optimization problem associated with the maxmin criterion is as follows.

$$\begin{array}{lll} \max_{\delta_1(t),\ldots,\delta_K(t)} & \min_{i\in[K]} & \frac{d_i(t-1)+\delta_i(t)}{\tau_i t} \\ \text{subject to} & & 0 \leq \delta_i(t) \leq w_i(t), \quad i \in [K] \\ & & \sum_{i=1}^{K} \delta_i(t) = \Delta(t) \\ & & \delta_i(t) \text{integer.} \end{array} \tag{6.15}$$

This problem can be reformulated as a mixed–integer programming problem as

$$\begin{array}{lll} \max & \beta & \\ \text{subject to} & \frac{d_i(t-1)+\delta_i(t)}{\tau_i t} - \beta \geq 0, & i \in [K] \\ & \sum_{i=1}^{K} \delta_i(t) = \Delta(t) & \\ & 0 \leq \delta_i(t) \leq w_i(t), & i \in [K] \\ & \delta_i(t) & \text{integer.} \end{array} \tag{6.16}$$

6.4 A greedy algorithm

Even though the optimization problems associated with the maxmin (or the maxmin) criteria are mixed integer programming problems, we prove that these problems can be solved by a greedy algorithm. In frame t, if $\Delta(t)$ packets are to be dropped (or M packets to be carried), then the greedy algorithm suggests that we could split the corresponding mixed integer programming problem into $\Delta(t)$ (or M) problems where exactly one cell needs to be dropped (or accepted for transmission). The decision in the latter problem is as simple as choosing a source, from among those that have at least one packet to drop (or at least one packet to offer), with the best (or worst) normalized QoS so far. The next packet would be dropped (or accepted for transmission) from the source chosen in the above step. Thus, the optimization problem can be decomposed into problems in which decisions are made on one packet at a time. Each one of these problems is such that the optimal solution has a simple structure and requires little computation.

In this section, we present a proof of the greedy algorithm for the MIP associated with the maxmin criterion. A strikingly similar proof can be constructed for the correctness of the greedy algorithm for the MIP associated with the minmax criterion. In problems (6.13) and (6.15), we can drop the t in the denominator in the objective function since every element is being divided by t. So the MIP in (6.16) can be reformulated as

$$\begin{array}{lll} \max & \beta & \\ \text{subject to} & \frac{d_i(t-1)+\delta_i(t)}{\tau_i} - \beta \geq 0, & i \in [K] \\ & \sum_{i=1}^{K} \delta_i(t) = \Delta(t) & \\ & 0 \leq \delta_i(t) \leq w_i(t), & i \in [K] \\ & \delta_i(t) & \text{integer.} \end{array} \tag{6.17}$$

Let $\Gamma(D)$ represent the parameterization of the problem when $\Delta(t) = D$. Since we are looking at decisions to be made in frame t, we could simplify the notation by denoting $d_i(t-1) = d_i, w_i(t) = w_i$ and so on. That is

$$\begin{array}{lll} \Gamma(D): & \max & \beta \\ & \text{subject to} & \frac{d_i+\delta_i}{\tau_i} - \beta \geq 0, \quad i \in [K] \\ & & \sum_{i\in[K]} \delta_i = D \\ & & 0 \leq \delta_i \leq w_i, \qquad i \in [K] \\ & & \delta_i \text{ integer.} \end{array} \tag{6.18}$$

and let $\beta^*(D)$ denote the value of the optimal solution of $\Gamma(D)$. It is clear that an optimal solution to $\Gamma(\Delta(t))$ is an optimal solution to the problem in (6.17).

Let

$$\begin{aligned} I_-(\mathbf{x}) &= \{i : x_i > 0\} \\ I_+(\mathbf{x}) &= \{i : x_i < w_i\}. \end{aligned} \tag{6.19}$$

Lemma 1 *$\beta^*(D)$ is non–decreasing in D*

Proof: Suppose $\exists$ D such that $\beta^*(D+1) < \beta^*(D)$. Let $\mathbf{x}^{(D)}$ denote the optimal solution for $\Gamma(D)$. Let $k \in I_+(\mathbf{x}^{(D)})$. Define $\mathbf{y}^{(D+1)}$ as

$$y_i^{(D+1)} = \begin{cases} x_i^{(D)} + 1, & i = k \\ x_i^{(D)}, & \text{otherwise.} \end{cases} \tag{6.20}$$

Since

$$\frac{d_i + x_i^{(D)}}{\tau_i} \geq \beta^*(D), \qquad i \in [K], \tag{6.21}$$

and $x_i^{(D)} \leq y_i^{(D+1)}, \qquad i \in [K]$, we have

$$\frac{d_i + y_i^{(D+1)}}{\tau_i} \geq \beta^*(D) > \beta^*(D+1). \tag{6.22}$$

Since $\mathbf{y}^{(D+1)}$ is feasible for $\Gamma(D+1)$, this contradicts the assumption that $\beta^*(D+1)$ is the optimal value of $\Gamma(D+1)$. $\square$

Lemma 2 *Let $\mathbf{x}^{(D)}$ denote the optimal solution for $\Gamma(D)$. Let*

$$j = \arg\min_{i\in[K]} \frac{d_i + x_i^{(D)}}{\tau_i}. \tag{6.23}$$

Then

$$\beta^*(D) \leq \beta^*(D+1) \leq \beta^*(D) + \frac{1}{\tau_j}. \tag{6.24}$$

Proof: From Lemma 1, we know that $\beta^*(D) \leq \beta^*(D+1)$. Suppose $\beta^*(D+1) > \beta^*(D) + \frac{1}{\tau_j}$. Let $\mathbf{z}^{(D+1)}$ denote the optimal solution for $\Gamma(D+1)$. Define $\mathbf{y}^{(D)}$ as

$$y_i^{(D)} = \begin{cases} z_i^{(D+1)} - 1, & i = j \\ z_i^{(D+1)}, & \text{otherwise.} \end{cases} \tag{6.25}$$

Since

$$\frac{d_i + z_i^{(D+1)}}{\tau_i} \geq \beta^*(D+1), \qquad i \in [K], \tag{6.26}$$

we get for $i \in [K]$

$$\frac{d_i + y_i^{(D)}}{\tau_i} \geq \beta^*(D+1) - \frac{1}{\tau_j} > \beta^*(D). \tag{6.27}$$

Since $\mathbf{y}^{(D)}$ is a feasible solution for $\Gamma(D)$, this contradicts the assumption that $\beta^*(D)$ is the optimal value of $\Gamma(D)$. $\square$

Lemma 3 *Assume $\mathbf{x}^{(D)}$ is optimal for $\Gamma(D)$ and*

$$j = \arg\min_{i \in [K]} \frac{d_i + x_i^{(D)}}{\tau_i}.$$

If $j \notin I_+(\mathbf{x}^{(D)})$, then

$$\beta^*(d) = \beta^*(D), d = D, D+1, \ldots \tag{6.28}$$

Proof: Let $d \geq D$. From Lemma 1, we know that

$$\beta^*(d) \geq \beta^*(D). \tag{6.29}$$

Any optimal solution, $\mathbf{z}^{(d)}$ for $\Gamma(d)$ should satisfy $z_j^{(d)} \leq w_j$. Since $j \notin I_+(\mathbf{x}^{(D)})$, this means $x_j^{(D)} = w_j$. Hence,

$$\begin{aligned} \beta^*(d) &\leq \tfrac{d_j + z_j^{(d)}}{\tau_j} \\ &\leq \tfrac{d_j + x_j^{(D)}}{\tau_j}, \quad \text{since } w_j = x_j^{(D)}, \\ &\leq \beta^*(D). \end{aligned} \tag{6.30}$$

From equations (6.29) and (6.30), we have

$$\beta^*(d) = \beta^*(D). \tag{6.31}$$

$\square$

Theorem 1 *Let* $\mathbf{x}^{(D)} = (x_1^{(D)}, \ldots, x_K^{(D)})$ *be an optimal solution for* $\Gamma(D)$. *Define*

$$\mathbf{x}^{(D+1)} = \mathbf{x}^{(D)} + \mathbf{e}_j \tag{6.32}$$

where

$$j = \arg \min_{i \in I_+(\mathbf{x}^{(D)})} \frac{d_i + x_i^{(D)}}{\tau_i}. \tag{6.33}$$

$\mathbf{x}^{(D+1)}$ *is an optimal solution for* $\Gamma(D+1)$.

Proof: Let

$$l = \arg \min_{i \in [K]} \frac{d_i + x_i^{(D)}}{\tau_i}. \tag{6.34}$$

Suppose $l \notin I_+(\mathbf{x}^{(D)})$, (*i.e.* $l \neq j$) then by Lemma 3, $\beta^*(D+1) = \beta^*(D)$. Any feasible solution for $\Gamma(D+1)$ that has $x_j^{(D)}$ as its j-th component is optimal. Hence, $\mathbf{x}^{(D+1)}$ is optimal for $\Gamma(D+1)$.

Henceforth, assume $l \in I_+(\mathbf{x}^{(D)})$, *i.e.* $l = j$. Let $\beta^*(D+1)$ denote the value of the objective function for $\Gamma(D)$. Let the corresponding optimal solution be $\mathbf{z}^{(D+1)}$.

Since $\mathbf{x}^{(D+1)}$, as defined in equation (6.32), is feasible for $\Gamma(D+1)$, the value of the objective function obtained by $\mathbf{x}^{(D+1)}$ is at most $\beta^*(D+1)$. That is

$$\min(\beta^*(D) + \frac{1}{\tau_j}, \min_{i \neq j} \frac{d_i + x_i^{(D)}}{\tau_i}) \leq \beta^*(D+1). \tag{6.35}$$

From Lemma 2, we have

$$\beta^*(D) + \frac{1}{\tau_j} \geq \beta^*(D+1). \tag{6.36}$$

In order to complete the proof, it is sufficient to show that

$$\min_{i \neq j} \frac{d_i + x_i^{(D)}}{\tau_i} \geq \beta^*(D+1). \tag{6.37}$$

If $\beta^*(D+1) = \beta^*(D)$, then it is easy to see that $\mathbf{x}^{(D+1)}$ is optimal since the relation in equation (6.37) is satisfied. Henceforth, we assume $\beta^*(D+1) > \beta^*(D)$. For any optimal solution, $\mathbf{z}^{(D+1)}$, for $\Gamma(D+1)$, we must have

$$z_j^{(D+1)} \geq x_j^{(D+1)} + 1. \tag{6.38}$$

Otherwise, $\beta^*(D+1) = \beta^*(D)$, which contradicts our assumption.

Suppose $\exists\ k \neq j$ such that

$$\frac{d_k + x_k^{(D)}}{\tau_k} < \beta^*(D+1). \tag{6.39}$$

From the above equation,

$$\frac{d_k + z_k^{(D+1)}}{\tau_k} \geq \beta^*(D+1) > \frac{d_k + x_k^{(D)}}{\tau_k}. \tag{6.40}$$

Thus, we have $z_k^{(D+1)} > x_k^{(D)}$. Since

$$1 + \sum_{i=1}^{K} x_i^{(D)} = \sum_{i=1}^{K} z_i^{(D+1)},$$

and from equation (6.38), $\exists\ n$ such that

$$x_n^{(D)} > z_n^{(D+1)}.$$

Also,

$$\frac{d_n + x_n^{(D)}}{\tau_n} > \frac{d_n + z_n^{(D+1)}}{\tau_n} \geq \beta^*(D+1) \geq \beta^*(D).$$

We consider the following two cases:

- Case 1: $\frac{d_k + x_k^{(D)}}{\tau_k} > \beta^*(D)$

  Define $\mathbf{y}^{(D)}$ as follows:

$$y_i^{(D)} = \begin{cases} x_n^{(D)} - 1, & i = n, \\ x_j^{(D)} + 1, & i = j, \\ x_i^{(D)}, & \text{otherwise.} \end{cases} \tag{6.41}$$

  $\mathbf{y}^{(D)}$ is feasible for $\Gamma(D)$, but yields an optimal value better than $\beta^*(D)$ and thus leads to a contradiction. Hence, the relation in equation (6.37) is true and $\mathbf{x}^{(D+1)}$ is an optimal solution for $\Gamma(D+1)$.

- Case 2: $\frac{d_k + x_k^{(D)}}{\tau_k} = \beta^*(D)$

  Define $\mathbf{y}^{(D)}$ as follows:

$$y_i^{(D)} = \begin{cases} x_n^{(D)} - 1, & i = n, \\ x_k^{(D)} + 1, & i = k, \\ x_i^{(D)}, & \text{otherwise.} \end{cases} \tag{6.42}$$

We must have $x_k^{(D)} < w_k$ to be able to construct $\mathbf{y}^{(D)}$ as above. Otherwise, from Lemma 3, we must have $\beta^*(D+1) = \beta^*(D)$. Thus, it is possible to construct another optimal solution, $\mathbf{y}^{(D)}$, for $\Gamma(D)$ in which

$$\frac{d_k + y_k^{(D)}}{\tau_k} > \beta^*(D).$$

We could do this for every k for which Case 2 is true. We could then apply Case 1 to show that $\mathbf{x}^{(D+1)}$, as defined by equation (6.32), is optimal for $\Gamma(D+1)$.

□

Theorem 1 proves that the problem $\Gamma(D+1)$ could be solved by first solving $\Gamma(D)$ and then using the optimal solution from $\Gamma(D)$ to solve another optimal solution $\Gamma'(1)$. If $\mathbf{x}^{(D)}$ is the optimal solution for $\Gamma(D)$, we could define another problem $\Gamma'(1)$ as follows:

$$\begin{array}{lll} \max\limits_{\delta'_1,\ldots,\delta'_K} & \min_{i\in[K]} & \dfrac{d_i + x_i^{(D)} + \delta'_i}{\tau_i} \\ \text{subject to} & \sum_{i=1}^{K} \delta'_i = 1 & \\ & 0 \le \delta'_i \le w_i - x_i^{(D)}, & i \in [K], \\ & \delta'_i & \text{integer.} \end{array} \tag{6.43}$$

Clearly, the optimal solution to this problem is given by equation (6.32). Working backwards from D, it is possible to solve $\Gamma(D)$ as D separate problems similar to $\Gamma'(1)$ given above.

6.5 Convergence of QoS

Assume one of the above–mentioned criteria is implemented to allocate bandwidth among the K sources in the system. In this section, we present two conjectures on the convergence of the normalized QoS achieved by each source. If these conjectures are assumed to be true, then we can derive an expression that determines the feasibility of the target QoS for each source under the given available bandwidth. This expression can be used for connection admission control purposes when a new source requests a connection to the system. The feasibility condition can then be evaluated to determine if the new source can be admitted into the system without violating the feasibility of the QoS targets for the sources already in the system. If it is possible still to satisfy every source's QoS target (including the new sources), then the source can be admitted. If not, we can determine what target QoS could be feasible for the new source and negotiate if the source wants to get admitted with a lower level of acceptable QoS (*i.e.* higher packet loss) than its initial specification.

The first conjecture, called the convergence conjecture, states that the normalized QoS values achieved by all the sources converge to some constant as t approaches ∞, as long as the long–term bit–rate of each source goes to a constant for each source. Thus, whatever the packet arrival process at each source is, if the long–term average converges to a constant, then the normalized QoS achieved by each source converges to a constant.

Conjecture 2 *If* $\forall\ i,\ \ \frac{n_i(t)}{t} \rightarrow \lambda_i,$ *as* $t \rightarrow \infty$, *then* $\forall\ i \in [K]$,

$$\frac{q_i(t)}{\tau_i} \rightarrow \alpha_i, \qquad as\ t \rightarrow \infty. \tag{6.44}$$

If $\alpha_i \leq 1, \forall i$, then the QoS target is being met for each source in the system. If, on the other hand, $\alpha_i > 1$ for a particular source i, then the QoS target for that source is not being satisfied. The normalized QoS achieved by this source i is higher than its target τ_i.

The second conjecture, called the fairness conjecture, states that if the condition in equation (6.45) is satisfied, then the normalized QoS achieved by each source converges to the same constant.

Conjecture 3 *If* $\forall\ i$,

$$E\left[w_i(t)|\sum_{j=1}^{K} w_j(t) > M\right] \geq \frac{\tau_i}{\sum_{j=1}^{K}\tau_j} E\left[\Delta(t)|\sum_{j=1}^{K} w_j(t) > M\right], \tag{6.45}$$

then $\forall\ i$,

$$\frac{q_i(t)}{\tau_i} \rightarrow \alpha, \qquad as\ t \rightarrow \infty. \tag{6.46}$$

Intuitively, the condition in equation (6.45) states that, on the average, the number of packets offered by each source should be at least its fraction of the total packet loss in all frames where there is packet loss. In Appendix B, we have illustrated an example where the condition in equation (6.45) is not satisfied and the normalized QoS values achieved by the sources do not converge to the same value.

Henceforth, in this discussion, we assume that the above two conjectures are true. As before, if $\alpha \leq 1$, then the given bandwidth is sufficient to satisfy the QoS target for each source in the system. If not, then all the sources face, relatively, the same degradation in service. If a source i has an acceptable packet loss value that is twice as much as that of source j, that is, source i is willing to accommodate twice as much loss as source j, then the packet loss for source i is, asymptotically, twice that of source j.

The total packet loss depends only on the superposition of the K packet arrival processes, $A_i = \{w_i(t), t = 1, 2, \ldots\}, i \in [K]$, and the number of available slots, M. It is independent of the maxmin/minmax criterion or any other criterion that we may use to determine how to distribute this packet loss among the K sources. Hence,

$$\frac{1}{t}\sum_{y=1}^{t} \Delta(y) \rightarrow E[\Delta(A_1, A_2, \ldots, A_K, M)], t \rightarrow \infty. \tag{6.47}$$

From the above equation and Conjecture 3, we get

$$\alpha \sum_{i=1}^{K} \tau_i = E[\Delta(A_1, A_2, \ldots, A_K, M)]. \tag{6.48}$$

Thus the condition for the feasibility of the QoS targets given the arrival process and the available bandwidth reduces to

$$\sum_{i=1}^{K} \tau_i \geq E[\Delta(A_1, A_2, \ldots, A_K, M)]. \tag{6.49}$$

All the information that we need is the quantity on the right hand side of equation (6.49) to determine the feasibility of the QoS targets.

Chapter 7

Conclusions

The next generation of wireless information networks are expected to support a wide range of services – voice, video, data from computer communications, to name a few – with stringent quality of service constraints. In order to support high user population with acceptable quality, high utilization of the scarce bandwidth that is available is required. One of the ways to achieve high throughput is by efficient allocation of the available bandwidth among the many competing users.

In this dissertation, we have addressed bandwidth allocation schemes in time–slotted integrated multiservices wireless networks. In particular, we have concentrated on studying bandwidth allocation schemes constrained to meet quality of service (QoS) requirements in two systems – one based on PRMA [19] and the other based on a wireless ATM system [54].

7.1 Frame Reservation Multiple Access (FRMA)

In PRMA, as proposed in [19], the base station transmits a feedback message at the end of every time slot. The mobile terminals are required to listen to this feedback message in every slot to update their knowledge of the system state. Thus, the receiver at the mobile terminal is switched on in every time slot. We have proposed modifications to PRMA to derive Frame Reservation Multiple Access (FRMA) with a view to reducing the receiver activity at the mobile terminal. Apart from leading to longer battery life (and hence lighter handsets), this permits the mobile station to identify base stations in neighboring cells that are potential candidates for a handoff (in case one is necessary). When combined with time–division duplex (TDD) operation (Figure 3.1), the base station is given sufficient time to process the information received in the uplink frame and generate a broadcast message.

We have developed an exact analytical method to compute the voice packet dropping probability (the performance measure of interest) in an FRMA system. To answer the interesting question of what is a good choice of permission probability in every frame, we have proposed four schemes and discuss the advantages and disadvantages of these schemes. Among the four schemes, the adaptive permission probability scheme, which chooses a value for the permission probability based on the number of reserved slots at the start of the frame, is easy to implement, does not place any computational burden other than a table look–up in every frame, and, importantly, does not sacrifice the simplicity of PRMA. For the adaptive permission probability scheme, we have formulated an optimization problem to choose permission probability values that minimize voice packet dropping probability. Comparison of the performance of this scheme to the constant permission probability scheme shows that the performance of the adaptive scheme was very close to that of the lower bound for the system parameters under consideration (Table 3.1). At higher loads, the performance of the adaptive scheme is closer to that of the constant scheme and this is also reflected in the nature of the optimal solutions to the adaptive scheme at higher load. The capacity of a PRMA system with the same operating parameters is equal to that achieved by a FRMA system using the constant permission probability scheme. But with the adaptive scheme, we are able to improve upon this capacity with the addition of one more voice conversation.

We have proposed a new scheme for the integration of voice and data in FRMA. The new scheme is based on a logical separation of the voice and data subsystems with a boundary that is moved dynamically to reflect the changing traffic load of the voice subsystem. The bandwidth allocated to the voice subsystem is again divided into traffic region (or the slots that carry voice information) and voice contention region. The size of the voice contention region is chosen dynamically to satisfy the QoS requirement of the voice users – a voice packet dropping probability of at most 0.01. In each frame, we have to choose the size of the voice contention region and the permission probability that the contending voice users will use to contend during that frame. We have formulated an optimization problem to choose these design parameters in such a way that the expected bandwidth available for the data subsystem is maximized while the voice packet dropping probability is maintained below 0.01. A heuristic method was used to solve this problem. Even though, the *optimal* solutions to this problem, obtained by the heuristic method, are approximate, the resultant performance metrics are exact for the particular choice of design parameters. We embedded a simple data protocol in the data subsystem and studied its performance using simulations. The results show that the sustainable data rate (Figure 4.8) is much higher than that achieved by earlier work [38, 62].

7.2 Centralized Medium Access Control

Chapter 5 contains a brief description of a proposed wireless ATM system [54]. In this system, the base station has complete knowledge of the bandwidth requirements of all the users in the system. The bandwidth allocation is thus centralized, in contrast to systems based on PRMA and its variants where the base station is less active in the allocation of bandwidth.

Motivated by the problem of bandwidth allocation for variable bit–rate (VBR) sources, we have studied centralized bandwidth allocation schemes with the sum of the peak rates of the sources exceeding the available capacity. Assuming that the packets have a limited lifetime of one frame (which is the also smallest unit of time over which the base station allocates bandwidth), we observe that some packet loss is inevitable. We have proposed two criteria that could be used to determine how to distribute this inevitable loss among the sources in a *even* manner. These two criteria each lead to a mixed–integer programming (MIP) problem. We have shown that a greedy algorithm can be used to solve these MIPs in a such a manner that it is sufficient to look at one dropped packet at a time and decide on which source would drop that packet. We have also conjectured that under some simple conditions, the ratio of the packet loss faced by each one of the sources in the system to the maximum packet loss (referred to as the target QoS) that each source considers acceptable could converge to a constant that is the same for each source. If the value of this constant is less than or equal to 1.0, then all the sources are getting an acceptable performance. Otherwise, they all face the same relative degradation. Assuming the conjectures are true, we can derive a simple relation that could be used to determine feasibility of the target QoS of the sources in the system using the given available capacity. This relation could be used in connection admission control (CAC) to decide whether to admit a new connection being attempted or deny it.

Appendix A

Conditional Distribution of the Number of Successful Terminals

Consider an arbitrary frame with R reserved slots, v voice contention slots, and C contending voice terminals. Let S denote the number of contending terminals that succeed in obtaining a reservation at the end of the frame. Let J denote the number of available slots in which there was a successful transmission. In this appendix, we show how to calculate the conditional distribution $P\{S = s|J = j\}$.

Let S_i denote the number of successful transmissions by user i $(i = 1, 2, \ldots, C)$. so that $J = S_1 + \cdots + S_C$. Given $J = j$, consider one of the j slots in which a success occurred. Since all users have the same permission probability, each of the C users has a conditional probability of success $1/C$ in such a slot. Moreover, because the outcome in each of these j slots is independent of that in all other slots, the conditional distribution of $S_1, \ldots, S_C$ given $J = j$ is the multinomial distribution

$$P\{S_1 = s_1, \ldots, S_C = s_C|J = j\} = \begin{cases} \dfrac{j!}{s_1! \cdots s_C!} \dfrac{1}{C^j}, & s_1 + \cdots + s_C = j, \\ 0, & \text{otherwise.} \end{cases} \tag{A.1}$$

Notice that $S = 0$ iff $J = 0$. Henceforth, we assume $J > 0$ and thus $S > 0$. Let A_s denote the set of all $(S_1, \ldots, S_C)$ for which $S = s$ so that

$$P\{S = s|J = j\} = \sum_{(s_1,\ldots,s_C) \in A_s} \frac{j!}{s_1! \ldots s_C!} \frac{1}{C^j} \qquad 1 \leq s \leq \min(j, C). \tag{A.2}$$

We observe that $S = s$ if there is a subset of exactly s users that have made at least one successful transmission. Let L_s denote the set of s element vectors $(m_1, \ldots, m_s)$ such that $\sum_{i=1}^{s} m_i = j$ and $m_i \geq 1$ for $1 \leq i \leq s$. Hence, $(s_1, \ldots, s_C) \in A_s$ if $s_1, \ldots, s_C$ contains a subsequence $(s_{l_1}, \ldots, s_{l_s}) \in L_s$ corresponding to the s users each with at least one successful transmission. Since there are $\binom{C}{s}$ possible subsets of s users,

$$P\{S=s|J=j\} = \binom{C}{s} \sum_{(s_{l_1},\ldots,s_{l_s})\in L_s} \frac{1}{C^j}\frac{j!}{s_{l_1}!\cdots s_{l_s}!}, \qquad 1 \le s \le \min(j,C). \tag{A.3}$$

We define

$$M_j(s) = \sum_{(m_1,\ldots,m_s)\in L_s} \frac{j!}{m_1!\cdots m_s!}, \qquad 1 \le s \le \min(j,C), \tag{A.4}$$

so that

$$P\{S=s|J=j\} = \binom{C}{s}\frac{M_j(s)}{C^j}. \tag{A.5}$$

We note that $M_j(s)$ represents the number of length j ordered sequences of s distinguishable objects such that each object appears at least once in each sequence. We can recursively calculate $M_j(s)$. First, we note that if each object is not required to appear, then the number of length j ordered sequences of s distinguishable objects is s^j. Hence, $M_j(s)$ equals s^j less the number of ordered length j sequences in which one or more objects do not appear. Since there are $\binom{s}{i}$ subsets of i objects and since there are $M_j(s-i)$ ordered sequences of $s-i$ objects in which all $s-i$ objects appear at least once, we have

$$M_j(s) = s^j - \sum_{i=1}^{s-1}\binom{s}{i} M_j(s-i). \tag{A.6}$$

Finally, since $M_j(1) = 1$, we have a recursive method for the computation of $M_j(s)$.

Appendix B

Example of centralized MAC

Consider two sources – one of them generating information at a constant rate while the other one is a bursty source. Let

$$w_1(t) = 1, t = 1, 2, \ldots \tag{B.1}$$

and

$$P\{w_2(t) = i\} = \begin{cases} 1 - 10^{-6}, & \text{if } i = 0, \\ 10^{-6}, & \text{if } i = 10^6, \\ 0, & \text{otherwise.} \end{cases} \tag{B.2}$$

Source 1 is a CBR source, while source 2 generates bursts of information with long idle periods in between. Let the number of available slots, M is 1. Also, let $\tau_1 = \tau_2 = 0.5$ Clearly,

$$P\{w_1(t) + w_2(t) > M\} = 10^{-6} \tag{B.3}$$

and $w_1(t) + w_2(t) > M$ if and only if $w_2(t) = 10^6$.

$$E[w_1(t)|w_1(t) + w_2(t) > M] = 1, \tag{B.4}$$

$$E[w_2(t)|w_1(t) + w_2(t) > M] = 10^6. \tag{B.5}$$

Also,

$$E[\Delta(t)|w_i(t) + w_2(t) > M] = 10^6. \tag{B.6}$$

The given values of τ_1 and τ_2 violate the condition in equation (6.45). Let us look at what the normalized QoS values would converge to. Whenever the bursty source has packets to offer to the system, it will experience a loss of at least $10^6 - 1$ packets. Irrespective of whether we implement the minmax or the maxmin criterion, we must have,

$$0 \le E[\delta_1(t)|w_1(t)+w_2(t)>M] \le 1,$$
$$10^6 - 1 \le E[\delta_2(t)|w_1(t)+w_2(t)>M] \le 10^6. \tag{B.7}$$

Since

$$E[\delta_i(t)|w_1(t)+w_2(t) \le M] = 0, i = 1,2, \tag{B.8}$$

we have

$$0 \le E[\delta_1(t)] \le 10^{-6},$$
$$1 - 10^{-6} \le E[\delta_2(t)] \le 1. \tag{B.9}$$

Since

$$q_i(t) \to E[\delta_i(t)], i = 1,2, \tag{B.10}$$

we cannot have convergence of $\frac{q_i(t)}{\tau_i}, i = 1,2$ to the same value (given the assumed values of τ_1 and τ_2).

On the other hand, consider $\tau_1 = 10^{-8}$ and $\tau_2 = 1$. Then the condition in equation (6.45) is satisfied. Assume that we are implementing the maxmim criterion to determine how to distribute the packet loss. Let us consider the frames where $w_2(t) > 0$. It is easy to see that out of every $10^8 + 1$ packets dropped, source 1 will drop exactly one packet, while source 2 drops the other 10^8. Hence their achieved normalized QoS converge to the same value, which is $\frac{1}{1+10^{-8}}$.

References

[1] N. Abramson. The ALOHA system - another alternative for computer communications. In *Proc. Fall Joint Comput. Conf., AFIPS Conf.*, page 37, 1970.

[2] A. S. Acampora and M. Naghgshineh. An architecture and methodology for mobile–executed handoff in cellular ATM networks. *IEEE JSAC*, 12(8):1365–1375, October 1994.

[3] N. Amitay. Resource auction multiple access in the cellular environment. *Electronics Letters*, 28(8), April 1992.

[4] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, 1992.

[5] G. Bianchi, F. Borgonovo, L. Fratta, L. Musumeci, and M. Zorzi. C–PRMA : the centralized packet reservation multiple access for local wireless communications. In *Proc. GLOBECOM 1994*, 1994.

[6] P. T. Brady. A model for on–off speech patterns in two–way conversation. *Bell Syst. Tech. Journal*, 48(7):2445–2472, September 1969.

[7] J. I. Capetanakis. The multiple access broadcast channel : Protocol and capacity considerations. *IEEE Trans. Inform. Theory*, 25:505–515, 1979.

[8] D. C. Cox. Personal communications – a viewpoint. *IEEE Communications Magazine*, pages 8–20, November 1990.

[9] D. C. Cox. Wireless network access for personal communications. *IEEE Communications Magazine*, pages 96–115, December 1992.

[10] D. C. Cox. Wireless personal communications : What is it? *IEEE Personal Communications*, pages 20–35, April 1995.

[11] A. Demers, S. Keshav, and S. Shenkar. Analysis and simulation of a fair queueing algorithm. *Internet Res. and Exper.*, 1, 1990.

[12] J. Dunlop. Packet access mechanisms for cellular radio. *Electronics and Communication Engineering Journal*, 5(3), June 1993.

[13] M. Eastwood, L. Hanzo, and J. C. S. Cheung. Packet reservation multiple access for wireless multimedia communications. *Electronics Letters*, 29(13):1178–1180, June 1993.

[14] D. D. Falconer, F. Adachi, and B. Gudmundson. Time division multiple access methods for wireless personal communications. *IEEE Communications Magazine*, 33(1):50–57, January 1995.

[15] R. Fantacci. Generalized error control techniques for integrated service packet networks. *IEEE Trans. Comm.*, 42(10):2815–2821, October 1994.

[16] R. R. Gejji. Mobile multimedia scenario using ATM and microcellular technologies. *IEEE Transactions on Vehicular Technology*, 43(3):699–703, August 1994.

[17] D. J. Goodman. Cellular packet communications. *IEEE Trans. Comm.*, 38(8):1272–1280, August 1990.

[18] D. J. Goodman. Trends in cellular and cordless communications. *IEEE Communications Magazine*, pages 31–40, June 1991.

[19] D. J. Goodman, R. A. Valenzeula, K. T. Gayliard, and B. Ramamurthi. Packet reservation multiple access for local wireless communications. *IEEE Trans. Commun.*, 37:885–890, August 1989.

[20] D. J. Goodman and S. X. Wei. Efficiency of packet reservation multiple access. *IEEE Trans. Veh. Tech.*, 40:584–598, August 1991.

[21] L. Greenstein. Microcells in personal communications systems. *IEEE Communications Magazine*, pages 76–88, December 1992.

[22] B. Hajek and T. Van Loon. Decentralized dynamic control of a multiaccess broadcast channel. *IEEE Trans. Automat. Control*, 27:559–569, 1982.

[23] H. Hashemi. The indoor radio propagation channel. *Proc. of the IEEE*, 81(7):943–968, July 1993.

[24] J. F. Hayes. An adaptive technique for local distribution. *IEEE Trans. Commun.*, COM-26, Aug. 1978.

[25] D. M. Himmelblau. *Applied Nonlinear Programming.* McGraw–Hill, New York, 1972.

[26] J. C. Huang and T. Berger. Delay analysis of the modified 0.487 contention resoulution algorithm. *IEEE Trans. Inform. Theory*, 31:264–273, 1985.

[27] I. M. Jacobs, R. Binder, and E. V. Hoversten. General purpose packet satellite networks. *Proc. IEEE*, 66, Nov. 1978.

[28] S. Jangi and L. F. Merakos. Performance analysis of reservation random access protocols for wireless access networks. *IEEE Trans. Communications*, 42(2):1223–1234, February 1994.

[29] L. Kleinrock. *Queueing Systems Vol 2 : Computer Applications.* Wiley, New York, 1976.

[30] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part I – carrier sense multiple access modes and their throughput–delay characteristics. *IEEE Trans. Commun.*, COM-23:1400–1416, December 1975.

[31] A. G. Konheim and B. Meister. Service in a loop system. *Journal of the ACM*, 19:92–108, Jan. 1972.

[32] S. S. Lam. Packet broadcast networks : A performance analysis of the R–ALOHA protocol. *IEEE Trans. Comput.*, C–29:596–603, 1980.

[33] I. M. Leslie, D. R. McAuley, and D. L. Tennenhouse. ATM everywhere? *IEEE Network*, pages 40–46, March 1993.

[34] G. P. McCormick. *Nonlinear Programming : Theory, Algorithms, and Applications.* Wiley, New York, 1983.

[35] K. S. Meier-Hellstern, G. P. Pollini, and D. J. Goodman. A wireless service for the ieee 802.6 metropolitan area network. In *Proc. IEEE GLOBECOM '91*, Phoenix, AZ, December 1991.

[36] N. M. Mitrou, Th. D. Orinos, and E. N. Protonotarios. A reservation multiple access protocol for microcellular mobile–communication systems. *IEEE Trans. Veh. Tech.*, 39(4):340–351, November 1990.

[37] J. Mosley and P. A. Humblet. A class of efficient contention resoulution algorithms for multiple access channels. *IEEE Trans. Comm.*, 33:145–151, 1985.

[38] S. Nanda. Analysis of packet reservation multiple access : Voice and data integration for wireless networks. In *Proc. GLOBECOMM '90*, pages 1984–1988, San Diego, CA, 1990.

[39] S. Nanda, D. J. Goodman, and U. Timor. Performance of PRMA : A packet voice protocol for cellular systems. *IEEE Trans Veh. Tech.*, 40:585–598, August 1991.

[40] P. Narasimhan and D. J. Goodman. PRMA system performance with acknowledgement delay. Technical Report 67, WINLAB, Rutgers University, Piscataway, NJ 08854, January 1994.

[41] P. Narasimhan, R. Yates, and D. J. Goodman. Analysis of frame reservation multiple access. In *Proc. International Conference on Universal Personal Communications, 1994*, San Diego, CA, 1994.

[42] P. Narasimhan, R. Yates, and D. J. Goodman. Performance analysis of frame reservation multiple access. Technical Report 80, WINLAB, Rutgers University, Piscataway, NJ 08854, July 1994.

[43] P. Narasimhan and R. D. Yates. A new protocol for the integration of voice and data over PRMA. Technical Report 96, WINLAB, Rutgers University, Piscataway, NJ 08855, 1995.

[44] P. Narasimhan and R. D. Yates. A new protocol for the integration of voice and data over PRMA. In *Proc. PIMRC '95*, Toronto, Canada, September 1995.

[45] P. Narasimhan and R. D. Yates. A new protocol for the integration of voice and data over PRMA. *IEEE JSAC*, 1996.

[46] A. K. Parekh and R. G. Gallager. A generalized process sharing approach to flow control in integrated services network : The single node case. *IEEE/ACM Trans. Networking*, 1(3):344–357, June 1993.

[47] A. K. Parekh and R. G. Gallager. A generalized process sharing approach to flow control in integrated services network : The multiple node case. *IEEE/ACM Trans. Networking*, 2(2):137–150, April 1994.

[48] H. Qi and R. Wyrwas. Markov analysis for PRMA performance study. In *Proc. Vehicular Technology Conference 94*, Stockholm, Sweden, June 1994.

[49] P. Rauber. Duplex schemes and receiver activity for packet reservation multiple access. Master's thesis, Rutgers, The State University of New Jersey, New Brunswick, NJ, 1994.

[50] D. Raychaudhuri. ATM–based transport architecture for multiservices wireless personal communication networks. In *Proc. ICC'94*, pages 317.1.1–7, 1994.

[51] D. Raychaudhuri. Next generation wireless personal communication networks. In *Proc. IEEE/City U Wireless Communications Symposium*, Hong Kong, January 1995.

[52] D. Raychaudhuri. Wireless ATM : An enabling technology for multimedia personal communications. In *Proc. Mobile Multimedia Communications Workshop (MoMuC-2)*, Bristol, UK, April 1995.

[53] D. Raychaudhuri, D. Reininger, M. Ott, and G. Welling. Multimedia processing and transport for the wireless personal terminal scenario. In *Proc. SPIE Conference on Visual Communication and Information Processing (VCIP'95)*, May 1995.

[54] D. Raychaudhuri and N. D. Wilson. ATM–based transport architecture for multiservices wireless personal communication networks. *IEEE Journal of Special Areas in Communication*, 12(8):1401–1414, October 1994.

[55] R. Rivest. Network control by bayesian broadcast. *IEEE Trans. Inf. Theory*, 1985.

[56] K. Sriram, P. K. Varshney, and J. G. Shantikumar. Discrete-time analysis of integrated voice/data multiplexers with and without speech activity detectors. *IEEE Journal on Selected Areas in Communications*, SAC-1(6):1124–1132, December 1983.

[57] S. Tasaka. Stability and performance analysis of the R-ALOHA packet broadcast system. *IEEE Trans. Comput.*, C-32:717–726, August 1983.

[58] F. A. Tobagi. Multiaccess protocols in packet communication systems. *IEEE Trans. Comm.*, COM-28(4):468–488, 1980.

[59] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part III – Polling and (dynamic) split channel reservation multiple access. *IEEE Trans. Commun.*, COM-24:832–845, Aut. 1976.

[60] J. Turner. New directions in communications or which way to the information age? *IEEE Communication Mag.*, 24:8–15, 1986.

[61] 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 of Special Areas in Communication*, 11(6):870–884, August 1993.

[62] W. C. Wong and D. J. Goodman. A packet reservation multiple access protocol for integrated speech and data transmission. *IEE Procedings – I*, 139(6), December 1992.

[63] G. Wu, K. Mukumoto, and A. Fukuda. Analysis of an integrated voice and data transmission system using packet reservation multiple access. *IEEE Trans. Veh. Tech.*, 43(2):289–297, May 1994.

[64] H. Xie and D. J. Goodman. PRMA system performance with a variable number of calls. Technical Report 36, WINLAB, Rutgers University, June 1991.

[65] H. Xie, P. Narasimhan, R. Yuan, and D. Raychaudhuri. Data link control protocols for wireless ATM access channels. In *Proc. International Conference on Universal Personal Communications (ICUPC) '95*, Tokyo, Japan, November 1995.

[66] R. Yuan, S. K. Biswas, and D. Raychaudhuri. A signaling and control architecture for mobility support in wireless ATM networks. In *WINLAB workshop*, Piscataway, NJ, 1995.

[67] R. A. Ziegler and G. P. Pollini. An experimental implementation of the PRMA protocol for wireless communication. Technical Report 53, WINLAB, Rutgers University, Piscataway, NJ 08854, March 1993.

Vita

Parthasarathy Narasimhan

1983-87 B. Tech. (Electrical Engineering), Indian Institute of Technology, Madras, India.

1989-90 Graduate Assistant, New Jersey Institute of Technology.

1989-90 M.S. in Computer Science, New Jersey Institute of Technology, Newark, NJ.

1990-93 Teaching Assistant, Rutgers Center for Operations Research (RUTCOR).

1993-95 Research Assistant, Wireless Information Network Laboratory (WINLAB).

1996 PhD in Operations Research