University of Alberta

On Multiple Access and Bandwidth Efficiency in Wireless Communication Systems

by

Hongjun Zhang 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Computing Science

Edmonton, Alberta
Fall 2002

University of Alberta

Library Release Form

Name of Author: Hongjun Zhang

Title of Thesis: On Multiple Access and Bandwidth Efficiency in Wireless Communication Systems

Degree: Doctor of Philosophy

Year this Degree Granted: 2002

Hongjun Zhang
154RH, Michener Park
Edmonton, AB
Canada, T6H4M4

Date: July 19, 2002

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **On Multiple Access and Bandwidth Efficiency in Wireless Communication Systems** submitted by Hongjun Zhang in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Pawel Gburzynski

Wlodek Dobosiewicz

Jack Tuszynski

Xiaobo Li

Janelle Harms

Ioanis Nikolaidis

Date: July 9, 2002

Abstract

Future wireless communications networks will be characterized by their broad bandwidth and a wide range of services that they will provide, including multimedia services. In this dissertation, we study various aspects of the air interfaces in cellular radio communications systems and examine the problem of efficient allocation of the available bandwidth to integrate a number of services with different QoS requirements into one system. We present two MAC layer protocols to address this problem in two types of cellular communications systems. One protocol is based on TDMA technology, and the other protocol is based on CDMA technology. We also present in this dissertation, a review of the current cellular radio communications systems, a review of MAC layer schemes proposed in recent academic studies, and a brief description of the simulation tools that we developed for studying the performance of MAC layer protocols in wireless communications systems.

Our simulation tools cover data traffic simulation, radio channel simulation, and air interface simulation. Three typical data traffic models have been implemented in the data traffic simulation part. They are the On-Off traffic model, Discrete Autoregressive Model and Long Range Dependent (self-similar) traffic model. The radio channel simulation part includes Free Space Channel, Slow Fading Channel, Rayleigh Fading Channel, Rician Fading Channel, and Nakagami/Rician Fading Channel. Both TDMA-based and CDMA-based air interfaces (mainly at the MAC layer) are modeled in the air interface simulation component.

The new TDMA-based scheme that we designed is intended for carrying traffic from multiple services in mobile environments, e.g., Personal Communication Systems (PCS). In contrast to other TDMA-based protocols for mobile applications, instead of trying to fit the offered traffic to the slot size, our solution adapts the slot size to the offered traffic. In consequence, as demonstrated by our performance studies, the proposed scheme is more flexible and incurs lower bandwidth overhead than other TDMA-based solutions.

In the CDMA-based scheme, we present a novel radio channel structure based on slotted CDMA technology. The proposed new CDMA-based scheme is intended for carrying traffic with diverse bandwidth/QoS requirements. The essence of our approach is a com-

bination of flexible slotting with allocation of multiple codes to high-bandwidth mobiles. As demonstrated by our performance studies, the proposed scheme efficiently integrates multiple traffic classes into an unified CDMA system. It is highly flexible and incurs low overheads for a wide range of realistic traffic conditions.

Acknowledgements

I would like to thank my supervisor, Prof. Pawel Gburzynski, for his valuable advice and encouragement throughout my study and research at the University of Alberta. He believed in my abilities from the very beginning when he supported my application to the Ph.D. program. This dissertation would not be possible without his recognition.

I thank the professors in the communication networks group, Janelle Harms, Mike MacGregor, and Ioanis Nikolaidis, for the excellent research resources they provided in the lab that I used for my research projects.

I thank the staff in the administrative services group, computer operations group, hardware support group, and instructional support group for their support to me as a student and researcher.

I would like to thank many graduate students in the department who have provided me with assistance during my years at University of Alberta.

Finally and most importantly, I would like to thank my parents and my wife for their support during my Ph.D. program.

Contents

1 Background 1

1.1 Applications of wireless communications 1

1.2 The air interface 4

1.2.1 Physical layer 5

1.2.2 MAC protocols and radio resource management 7

1.3 A review of wireless MAC protocols 8

1.3.1 TDMA-based MAC protocols 9

1.3.2 CDMA-based MAC Protocols 14

2 DS-TDMA/CP 21

2.1 The protocol goals 21

2.2 Silent periods in CBR traffic 22

2.3 DS-TDMA/CP Protocol Description 23

2.3.1 Frame structure 23

2.3.2 Principles of bandwidth allocation 24

2.3.3 Bandwidth scheduling 25

2.4 Algorithms in DS-TDMA/CP 29

2.4.1 Contention resolution 29

2.4.2 Transmission scheduling 30

2.4.3 Processing at the mobile station 37

2.4.4 Scheduling UBR traffic 38

2.5 Virtual implementation 39

2.5.1 Traffic models 39

2.5.2 Numerical parameters of the model 42

2.6 Performance 44

2.6.1 QoS trade-offs 45

2.6.2 Burst responsiveness . . . 52
2.6.3 Overhead . . . 54
2.7 Conclusions . . . 56

3 BRICS **59**
3.1 The protocol goals . . . 60
3.2 Protocol prerequisites . . . 60
3.2.1 A multi-code transmitter for BRICS . . . 60
3.2.2 A multi-code receiver for BRICS . . . 62
3.3 BRICS: Protocol Description . . . 66
3.3.1 Logical channels . . . 66
3.3.2 Medium access . . . 70
3.3.3 Sessions . . . 72
3.3.4 Bandwidth allocation and admission control . . . 73
3.4 A sample configuration of the protocol . . . 76
3.4.1 The radio channels . . . 76
3.4.2 Contention resolution . . . 77
3.4.3 Signaling channels . . . 77
3.4.4 The voice traffic . . . 79
3.4.5 The video traffic . . . 82
3.4.6 The file transfers . . . 85
3.4.7 The short message service . . . 89
3.5 Performance . . . 91
3.5.1 The other protocols . . . 91
3.5.2 Bandwidth utilization . . . 92
3.5.3 Quality of service . . . 102
3.6 Conclusions . . . 117

4 Future Work **121**

A Methodology and Tools **123**
A.1 Methodology . . . 123
A.2 Simulation tools . . . 124
A.3 Traffic models . . . 125
A.3.1 The On-Off model . . . 125

A.3.2 The discrete autoregressive model (1) 126
A.3.3 The self-similar traffic model 128
A.4 Simulating the radio channel 129
A.4.1 Free space propagation 129
A.4.2 Slow fading 130
A.4.3 Multipath fading - Rayleigh fading and Rician fading 131
A.4.4 Nakagami fading 133
A.5 Modeling the air interface 135
A.5.1 The performance monitor 136
A.5.2 The base station 137
A.5.3 Generic mobile station 139
A.5.4 The system creator 142
A.5.5 The main function 143
A.5.6 Simulation Methodology 144

Abbreviations **145**

Bibliography **149**

List of Tables

2.1 Traffic priorities 30
2.2 Network parameters 43
2.3 Bandwidth overhead for 1Mb/s channel 55
2.4 Bandwidth overhead for different channel rates 56

3.1 Parameters of the voice model 79
3.2 Parameters of the video model 82
3.3 Parameters of the file transfer model 87
3.4 Parameters of the SMS model 90

A.1 The optimum values of the Rice parameter K_{dB} 133

List of Figures

1.1 Examples of protocol stacks for air interface 5
1.2 A typical frame structure in TDMA-based systems 9
1.3 Layouts of typical CDMA-based protocols 15
1.4 RACH structure in UTRA 17
1.5 A sample code tree 17

2.1 Uplink frame structure in DS-TDMA/CP 24
2.2 Structure of the frame announcement message 31
2.3 The CBR model 40
2.4 The VBR model 41
2.5 The UBR model 42
2.6 Bandwidth utilization in DS-TDMA/CP 45
2.7 CBR blocking rate in DS-TDMA/CP 46
2.8 Bandwidth utilization in DS-TDMA/CP, $C = 0.2, 0.6, 1$ 47
2.9 Bandwidth utilization in DS-TDMA/CP under different silent/active ratios 48
2.10 Bandwidth utilization in D-TDMA 49
2.11 Bandwidth utilization in DQRUMA 50
2.12 Blocking rate in DS-TDMA/CP for CBR traffic under constant CBR load . 51
2.13 Bandwidth utilization in DS-TDMA/CP under constant UBR load 52
2.14 Bandwidth utilization in DS-TDMA/CP under constant CBR load 53
2.15 Bandwidth utilization in DS-TDMA/CP under constant VBR load 54
2.16 Bandwidth utilization in DQRUMA under constant CBR load 55
2.17 Burst response of DS-TDMA/CP, short talkspurt, heavy load 56
2.18 Burst response of D-TDMA, heavy load 57
2.19 Burst response of DQRUMA, heavy load 57
2.20 Burst response of DS-TDMA/CP, long talkspurt, heavy load 58

3.1 MC-CDMA transmitter 61

3.2 MC-CDMA receiver 62
3.3 A PN code acquisition system based on parallel matched filters 63
3.4 Matched filter and correlator in a PN code acquisition system 64
3.5 BRICS frame structure 67
3.6 The video model 83
3.7 The file transfer model 86
3.8 The SMS model 89
3.9 Bandwidth utilization in BRICS ($W = 20\,\text{MHz}$) 93
3.10 Bandwidth utilization in BRICS ($W = 5\,\text{MHz}$) 94
3.11 Bandwidth utilization in WISPER ($W = 20\,\text{MHz}$) 95
3.12 Bandwidth utilization in WISPER ($W = 5\,\text{MHz}$) 96
3.13 Failure rate of access requests in WISPER 96
3.14 Packet drop rates in WISPER 97
3.15 Bandwidth utilization in VSG-CDMA ($W = 20\,\text{MHz}$) 98
3.16 Transmission probability distribution in VSG-CDMA ($W = 20\,\text{MHz}$) 100
3.17 Transmission failure rate in VSG-CDMA 101
3.18 Bandwidth utilization in S-CDMA ($W = 20\,\text{MHz}$) 101
3.19 Bandwidth utilization in BRICS under fixed voice and video load ($W = 20\,\text{MHz}$) 103
3.20 Voice and video block-rate in BRICS under fixed voice and video load . . . 104
3.21 Bandwidth utilization in BRICS under fixed voice and video load ($W = 10\,\text{MHz}$) 105
3.22 Bandwidth utilization in BRICS under fixed voice and video load ($W = 5\,\text{MHz}$) 106
3.23 Access request failure rate in BRICS under fixed voice and video load . . . 107
3.24 Fixed-load voice bandwidth utilization load in BRICS 108
3.25 Fixed-load video bandwidth utilization in BRICS 109
3.26 Fixed-load file transfer bandwidth utilization in BRICS 109
3.27 Fixed-load SMS bandwidth utilization in BRICS 110
3.28 Fixed-load voice bandwidth utilization in WISPER 110
3.29 Fixed-load video bandwidth utilization in WISPER 111
3.30 Fixed-load file transfer bandwidth utilization in WISPER 111
3.31 Fixed-load SMS bandwidth utilization in WISPER 112
3.32 Fixed-load SMS bandwidth utilization in VSG-CDMA 113
3.33 Fixed-load file transfer bandwidth utilization in VSG-CDMA 114

3.34 Transmission probability distribution under fixed file transfer load in VSG-CDMA 115
3.35 Transmission fail rate in VSG-CDMA under fixed file transfer load 116
3.36 Fixed-load video bandwidth utilization in VSG-CDMA 117
3.37 Transmission probability distribution in VSG-CDMA ($W = 5\,\text{MHz}$, fixed video load) 118
3.38 Transmission failure rate in VSG-CDMA under fixed video load 119
3.39 Fixed-load voice bandwidth utilization in VSG-CDMA 119
3.40 Transmission failure rate in VSG-CDMA under fixed voice load 120
3.41 Fixed-load bandwidth utilization of SMS S-CDMA ($W = 20\,\text{MHz}$) 120

A.1 The simulator structure 136
A.2 Bandwidth Usage Stabilization Time under DS-TDMA/CP 144

Chapter 1

Background

1.1 Applications of wireless communications

Cellular radio communication systems are wireless networks build of static base stations communicating with dynamic mobile stations. The system service area is divided into cells, ideally with no gaps or overlaps, with each cell being serviced by a separate base station. The cellular mobile communication systems, or just cellular systems, have experienced two generations and are advancing to the third generation. The first generation systems were analog and they provided basic voice service. Examples of the first generation (1G) systems [1, p.2] are AMPS (Advanced Mobile Phone System) in USA, TACS (Total Access Cellular System) in UK and NAMTS (Nippon Advanced Mobile Telephone System) in Japan. The second generation (2G) systems are digital. They are more spectrally efficient and offer a higher capacity than their predecessors. They provide better speech quality and make simple data service available to their customers. Examples of the second generation cellular systems [2, p.500-533] are GSM (Global System for Mobile Communications) and IS-95 (Interim Standard 95). GSM provides telephone services, data services, and supplementary ISDN services. The telephone services include standard mobile telephony and emergency access, the data services support facsimile and low rate (less than 9.6kbps) packet-switched traffic, and the supplementary ISDN services include call diversion, closed user groups, caller identification, and short message service (SMS), which allows GSM subscribers and base stations to transmit alphanumeric pages of limited length. IS-95 based mobile systems provide variable rate voice service and data services [3, 4], with the data services including circuit-switched asynchronous and packet data at a raw rate of up to 9.6kbps or 14.4kbps. Example applications are facsimile, simple asynchronous connections to host computers, Internet access, and short transaction applications (like credit card transactions).

In recent years, the demands for mobile communication services have been growing at an

explosive speed—from the basic services, e.g., voice, facsimile, and low-bit-rate (far less than 64kb/s) data, to a variety of wideband services, e.g., high-speed Internet access, video/high-quality image transmission, and e-commerce. These services create a heavy demand on the wireless bandwidth and have stimulated the development of the next generation mobile communication systems. The evolved second generation (2.5G) cellular mobile communication system—GPRS (General Packet Radio Service)—offers end-to-end packet switched data transfer at a rate up to 171.2kbps. In practice, because of sharing the radio bandwidth by multiple users, a much lower bit rate is available to an individual subscriber (approximately 40kbps on the average—[5]). The current GPRS systems support point-to-point (PTP) service that transfers data packets between two users. The service is offered in two models: PTP connection-less network service (e.g., for IP), and PTP connection-oriented network service (e.g., for X.25). Also, SMS messages can be sent over GPRS. Some supplementary services are planned, e.g., call forwarding unconditional (CFU), call forwarding on mobile subscriber not reachable (CFNR), and closed user group (CUG). Additional non-standardized services [5] may also be offered by GPRS service providers, such as access to data bases, messaging services, and tele-action services (e.g., credit card validations, lottery transactions, and electronic monitoring and surveillance systems).

Although 2.5G mobile systems, e.g., GPRS, provide higher bit rate packet switched network services, some applications, e.g., streaming video, are still seriously limited by the current bit rates. To this end, high speed data (HSD) mobile services will be supported in the third generation (3G) cellular systems. A number of third-generation mobile system standards have been rolled out (e.g., IMT-2000 [6] and UMTS [7]) and they are continually being improved to accommodate new developments in radio communication systems [8]. According to these standards, the third generation mobile systems should be able to offer at least 144kb/s (preferably 384kb/s) for high-mobility users with wide area coverage and 2Mb/s for low mobility users with local coverage. The third-generation cellular radio systems will provide basic and enhanced voice services, low-data-rate messaging services (e.g., e-mail, facsimile), medium-data-rate services (e.g., file transfer and Web browsing at rates of order 64-144kb/s), high-data-rate services (e.g., high-speed packet and circuit-based network access, high-quality video conferencing, and networked computing applications at rates between 64kb/s and 2Mb/s) [4]. Moreover, the third-generation systems will offer multimedia services, providing concurrent video, audio, and data sessions to support advanced interactive applications.

For this range of services, the third generation systems will have to define session types

meeting different quality of service (QoS) requirements. In the first release of UTMS, ATM Adaptation Layer 2 (AAL2) was chosen as the transport technology in the UTMS terrestrial radio access network (UTRAN). Consequently, the traffic classification of ATM service categories [9] has become the obvious choice for describing the QoS parameters of wireless sessions. Thus, the traffic classes include the following standard categories: Constant Bit Rate (CBR), Real-Time Variable Bit Rate (rt-VBR), Non-Real-Time Variable Bit Rate (nrt-VBR), Available Bit Rate (ABR), and Unspecified Bit Rate (UBR). The corresponding session parameters are [9]:

- Peak Cell Rate (PCR)
- Sustainable Cell Rate (SCR)
- Maximum Burst Size (MBS)
- Minimum Cell Rate (MCR)
- Cell Delay Variation (CDV)
- Maximum Cell Transfer Delay (Max CTD)
- Cell Loss Ratio (CLR)

With the evolution of the core networks in mobile telecommunications toward IP technology, another definition of traffic classes and associated QoS parameters has been proposed [10, 8]. Four traffic classes are defined according to their real-time service requirements, i.e., the conversational class (e.g., voice), the streaming class (e.g., streaming video), the interactive class (e.g., Web browsing), and the background class (e.g., downloading/uploading of electronic mail). With this classification, the corresponding session parameters are:

- Maximum Bit Rate
- Guaranteed Bit Rate
- Delivery Order of Service Data Units (SDUs) (in-sequence or not)
- Maximum SDU size
- SDU format requirements
- SDU error ratio

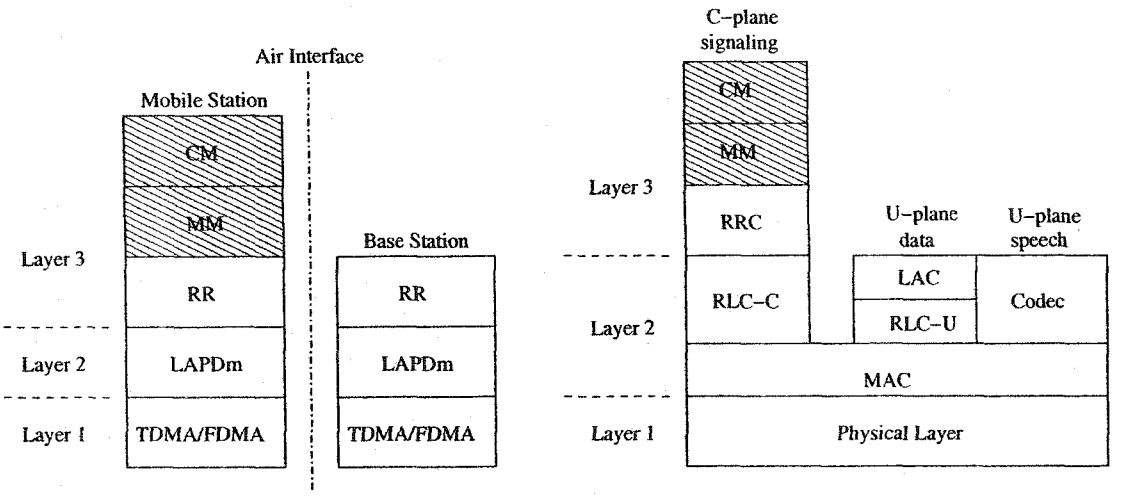
- Transfer delay
- Transfer handling priority
- Allocation/retention priority

As we can see, the QoS requirements of packet data applications in a mobile environment are quite diverse. The management of guaranteed end-to-end QoS services includes various aspects, which are additionally complicated by the inherent physical properties of the radio channels, notably the relatively high, variable, and unpredictable bit error rate. In a layered service architecture model [8], the end-to-end bearer service provided by the third generation UMTS systems can be decomposed into three main components: the Terminal Equipment/Mobile Terminal (TE/MT) local bearer service, the external local bearer service, and the UMTS bearer service. The TE/MT local bearer service supports communication between the components of a mobile station, which is responsible for the physical connection to the UTRAN system through the air interface. The external bearer service interfaces the UMTS core network with the destination node possibly located in an external network. The UMTS bearer service builds upon the radio bearer service (RAB) and the core network bearer service (CN). The role of the CN service is to efficiently control and utilize the core network. The RAB service supports the transport of signaling and user data through the air interface. This air interface is the primary topic of our present study.

1.2 The air interface

In a cellular mobile radio system, the digital data (or analog signals) are transmitted on a radio carrier between user terminals (mobile stations) and system access points (base stations). The interface between the mobile station and the base station is called the air interface. The air interface is a very important component of a cellular radio system. It determines the fundamental capacity of the system and its spectral efficiency [1].

The air interface is composed of both hardware and software. Generally, the first (bottom) three layers of the OSI model contribute to the air interface. For example, the GSM air interface protocol stack [11] is shown in Figure 1.1 (a), and the proposed air interface protocol stack for ETSI WCDMA [12] is depicted in Figure 1.1 (b). The shadowed areas in Figure 1.1 represent components belonging to the core network. In the GSM protocol stack, TDMA/FDMA technologies are applied in the physical layer, *LAPDm* is the data link layer protocol (which is a modified version of the LAPD protocol used in ISDN), *RR*



(a) GSM Air Interface Protocol Stack

(b) A Air Interface Procotol Stack Proposal for ETSI WCDMA

Figure 1.1: Examples of protocol stacks for air interface

is the radio resource management layer, *MM* is the mobility management layer, and *CM* is the connection management layer. In the ETSI WCDMA air interface, WCDMA is used in the physical layer.[1] *RLC* represents the radio link control protocols (RLC-C for the control plane and RLC-U for the user plane), *LAC* is the link access control protocol, *Codec* is the voice encoding/decoding component, *MAC* is the medium access control protocol, *RRC* is the radio resource control protocol (equivalent to *RR* in GSM), *MM* and *CM* having the same role as in the GSM model. Layer 2 in this air interface includes *MAC*, *RLC*, and *LAC*; *Codec* can either belong to layer 2 or layer 3.

1.2.1 Physical layer

Wireless communications use electromagnetic airwaves to transfer information from one point to another without relying on any structurally fixed connection. Radio waves are often referred to as radio carriers because they perform the function of transporting energy from a transmitter to a remote receiver. The data being transmitted are superimposed on the radio carrier through various modulation techniques, and they are extracted at the receiving end by the corresponding demodulation techniques. Since the frequency or bit rate of the modulated information is added to the carrier, the radio signal after modulation occupies a certain band of radio frequency. Multiple radio carriers can exist in the same space at the same time without interfering with each other, if the radio waves are transmitted on different frequency bands. From the viewpoint of the width and location of the electro-

[1] In the latest release UMTS R00 [8], a GPRS/EDGE radio access network is added.

magnetic spectrum used to support a wireless session, the main technologies for wireless communication fall into one of the following categories: narrowband, spread spectrum, and infrared. In a narrowband system, a single session occupies a tight and rigidly allocated range of radio frequencies. With the spread spectrum technology, the transmitted signal is spread over a wide range of frequencies. It therefore avoids concentrating power in a single narrow frequency band. There two main alternatives are Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS). The infrared (IR) systems use very high frequencies in the electromagnetic spectrum—just below visible light. Like light, IR signals cannot penetrate opaque objects. Infrared transmission can be based on either directed (line-of-sight) or diffuse technology.

The most promising of those generic approaches is DSSS, especially from the viewpoint of its efficiency in the third generation mobile systems. In particular, the spread-spectrum CDMA technology offers high spectrum efficiency, flexible (soft) capacity (or graceful degradation), multipath-resistance, and inherent frequency diversity [13, 14].

From the viewpoint of multiple access to the radio medium, three major classes of multiplexing strategies are employed in the physical layer: frequency division multiple access (FDMA), time division multiple access (TDMA), and code division multiple access (CDMA). FDMA divides the frequency band into a number of sub-bands, called *channels*, which are portions of the whole spectrum centered on different carrier frequencies. Each user is allocated an unique frequency band (channel), which cannot be shared with other users. The bandwidth of a single FDMA channel is typically narrow, i.e., FDMA is usually implemented in narrowband systems. The first generation mobile systems, e.g., AMPS, used FDMA. TDMA divides a frequency channel into a number of time slots, each of which is viewed as a sub-channel allocated to one user. In TDMA, a single carrier frequency is shared by several users with different users scheduled to transmit/receive at different times. Thus, data transmission for users of a TDMA system is not continuous. TDMA is used in second generation mobile systems, which use digital modulation to transmit both voice and data, e.g., GSM. CDMA separates different communication channels with different *spreading codes*. All users in a CDMA system use the same band of carrier frequency, yet they may transmit simultaneously without damaging each other's data. Before transmission, the signal is spread by using a specific code allocated to the user. At the receiver end, the data are extracted through despreading with the same code that was used for transmission. Theoretically, if the spreading codes are *orthogonal*, different channels can always be perfectly separated. However, under real-life conditions, the code orthogonality is degraded

because of transmission delays, distortions and interference. The number of simultaneous transmission on the same channel is limited, but the capacity limit is soft. This means that the system's performance gradually degrades as the number of simultaneous channels increases, and improves as the number of users decreases. Because all users share the same frequency, a CDMA system is a self-interference system, and elaborate power control schemes are needed to overcome the near-far problem. CDMA is used in second generation mobile systems, e.g., IS-95. The standard bodies currently focus on TDMA and CDMA techniques for the third generation systems [15].

1.2.2 MAC protocols and radio resource management

In a cellular radio communication system, the communication through the air interface involves two directions: *downlink* (from the base to the mobile station) and *uplink* (from the mobile to the base). In the downlink direction, the radio resources are used solely by base station—in a broadcast manner. In contrast, in the uplink direction, the radio resources must be shared among multiple mobile stations With FDMA or TDMA, a collision will happen if two or more users transmit at the same time and on the same frequency. In a CDMA system, signal blocking will occur if the number of simultaneous transmission is too high. As the available electromagnetic spectrum is limited, the radio resources must be utilized as efficiently as possible. This part is controlled by medium access control (MAC) protocols.

A medium access control protocol defines a set of rules that moderate access to the shared radio resources in an orderly and efficient manner. In wireless radio communication, MAC protocols are tightly coupled to specific multiplexing strategies in the physical layer. From this point of view, we have FDMA-based MAC protocols, TDMA-based MAC protocols, CDMA-based MAC protocols, and some hybrid MAC protocols based on combinations involving FDMA, TDMA, and/or CDMA. FDMA-based MAC protocols define rules for allocating radio frequency channels, TDMA-based MAC protocols define scheduling rules for allocating time slots, and CDMA-based access protocols define rules for allocating codes and transmission energy. Depending on which multiplexing techniques are involved, hybrid MAC protocols are usually designed for a set of basic resources, e.g., radio frequency channels, time slots, and transmission energy.

The MAC layer is usually a sub-layer of the radio access network layer 2 (*Data Link*). It receives services from the physical layer of transport channels and provides services to its upper sub-layer (e.g., RLC) via logical channels. Usually, a radio access MAC protocol

offers to its up-layer the *data transmission* service, which provides for data transfer between MAC peer entities in unacknowledged mode, *reallocation of radio resources and MAC layer parameters* (based on the request from its up-layer, e.g., RLC), and *traffic and quality measurements.* The latter may include a series of local measurements on traffic volume and signal quality parameters, whose results are sent to the up-layer for the purpose of controlling the radio resource usage.

If the radio communication system provides packet-switching services, e.g., as UTMS or GSM/GPRS, there is usually a radio link control (RLC) layer directly above the MAC layer. The RLC is a sub-layer of Layer 2 in the OSI model. It receives services from the MAC layer and provides services to its upper layers. These services include establishing a reliable radio link between the mobile station and base station. Usually, they cover data transmission (e.g., *transparent data transmission*, *unacknowledged data transmission*, and *acknowledged data transmission.*), *segmentation and re-assembly* of protocol data units (PDU), *concatenation/padding* of PDUs, and *error correction* by the appropriate algorithm (e.g., Selective Repeat, Go-Back-N, ARQ) [16].

Another function within the air interface is radio resource management. It mainly performs radio resource allocation in the unit provided by low layers (RLC, MAC, and physical layer). Depending on the transmission technology used in physical layer, radio resource management may have other functions, e.g., power control in CDMA systems.

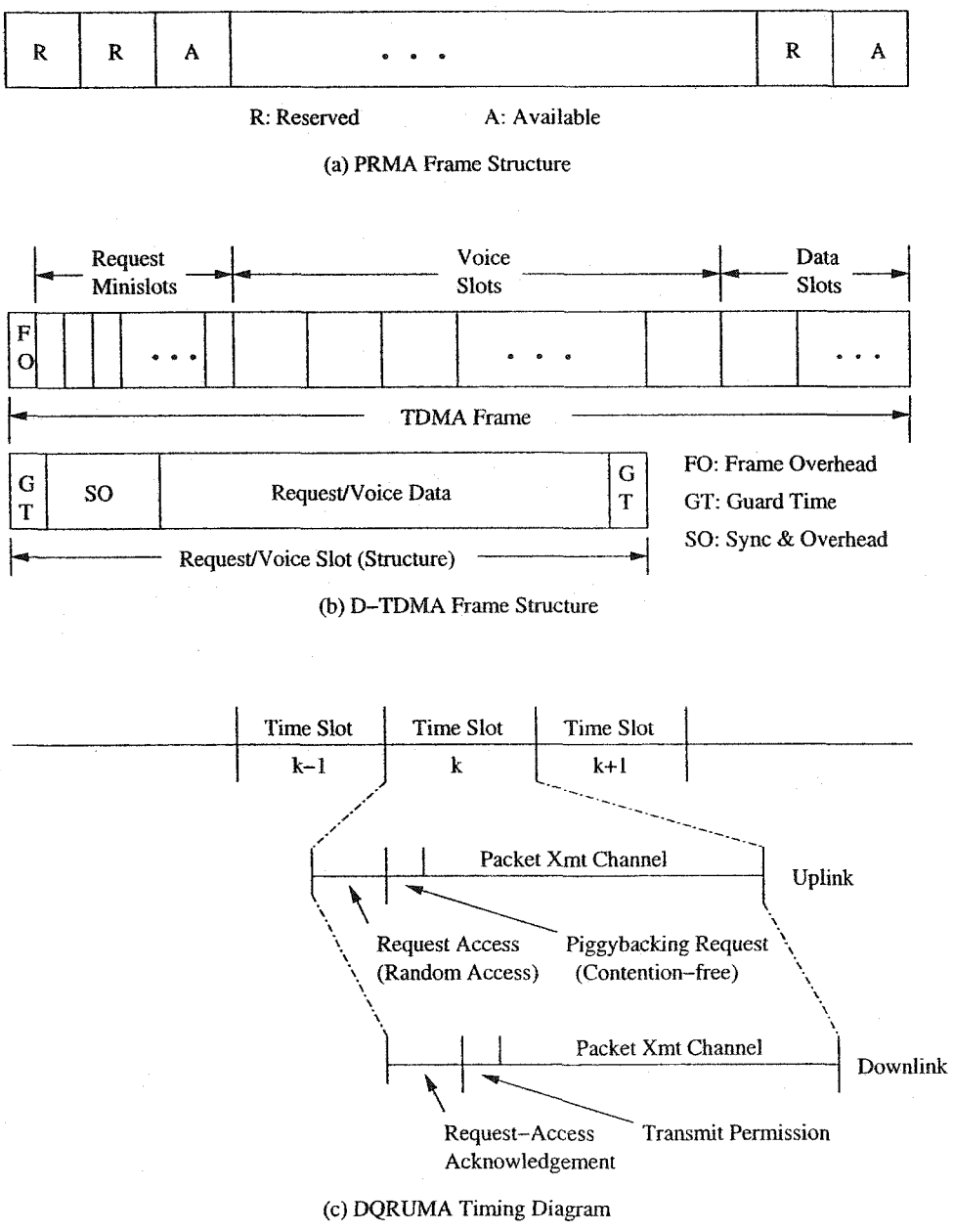
To provide broadband multimedia services within a radio communication system, more spectrally efficient modulation and medium access control techniques are needed. Especially, the medium access control schemes must be flexible to support a variety of differentiated services, including those that are well defined at present, as well as those that will emerge in the future. These issues are addressed in our work.

1.3 A review of wireless MAC protocols

To meet the requirements of 3G wireless communication systems, besides providing high speed radio links in the physical layer, another very serious challenge is to design a network architecture and an efficient and robust medium-access control (MAC) protocol that can integrate heterogeneous traffic types and meet their quality of service requirements. A number of MAC schemes have been proposed in recent years for integrating multiple traffic classes into a single system. Most research studies and standardization activities are focused on CDMA and TDMA technologies. In the following two subsections, we give a brief review of MAC schemes based on TDMA and CDMA.

1.3.1 TDMA-based MAC protocols

TDMA technology is well known at present because it has been deployed in several popular second generation wireless systems. Many TDMA-based medium access control protocols (some of them catering to various multimedia services) have been proposed for mobile networks, e.g., PRMA [17], DPRMA [18], C-PRMA [19], DRMA [20], D-TDMA [21, 22], RAMA [23], DQRUMA[24].



(a) PRMA Frame Structure

(b) D-TDMA Frame Structure

(c) DQRUMA Timing Diagram

Figure 1.2: A typical frame structure in TDMA-based systems

1.3.1.1 Packet Reservation Multiple Access (PRMA)

The simple frame structure of PRMA, shown in Figure 1.2(a), consists of a number of fixed-length slots that can be either "reserved" or "available." This status is indicated by a binary flag, called an acknowledgment, transmitted by the base station at the end of every slot. The fixed slot size and the number of slots per frame are tuned to efficiently accommodate periodic isochronous traffic of a constant bit rate (CBR) corresponding to a voice session. This boils down to making sure that one voice source needs precisely one slot per frame to expedite a talkspurt at its arrival rate. Mobile stations having packets to transmit compete for available slots using a modified ALOHA protocol. As soon as a CBR source manages to transmit a packet within a slot, the base station marks that slot as "reserved". Such a slot will remain reserved for the same periodic source in subsequent frames, until the mobile runs out of its burst and releases the slot by leaving it empty. Stations with non-CBR traffic (e.g., data) contend for available slots using the same ALOHA protocol. However, in contrast to CBR sources, they never reserve slots for more than one transmission. A slot acquired by a non-CBR source remains marked as "available," so it will be open for contention in the next frame.

PRMA differentiates between two classes of traffic and gives preferred treatment to CBR traffic for which isochronous service is critical. Its primary disadvantage is that the same slots are used for contention and for transmitting actual packets. Collisions are detected by the base station at the beginning of the affected slots, yet the remainders of those slots are unusable. Although this waste may be acceptable from the viewpoint of voice sessions consisting of multipacket bursts, it drastically reduces the bandwidth available to lower priority traffic. In essence, the behavior of PRMA for non-CBR traffic boils down to pure ALOHA, whose performance is known to be poor. Although the variant of ALOHA used in PRMA offers a somewhat better performance than pure ALOHA—because it accounts for the capture effect, whereby in some circumstances one of the colliding stations may transmit successfully—this improvement is rather insignificant. The colliding data packets waste precious slots that could be used by voice packets (new bursts). This reduces the overall useful bandwidth and affects the quality of service for the preferred traffic type. One has to conclude that PRMA was predominantly designed for CBR traffic, and that its treatment of other traffic types leaves a lot of room for improvement. By putting all non-CBR classes into the same basket, the protocol handles them all according to the same "best effort" delivery scheme, regardless of their expectations. For example, variable bit

rate (VBR) traffic with deadlines, typically resulting from compressed video sessions, is not well serviced with such treatment. Ideally, such traffic should not compete for service with sessions that require no hard deadlines (e.g., file transfers). This is a common flaw of many medium access schemes for cellular systems: they assume that the CBR traffic has absolute priority over everything else, and do not attempt to service other traffic classes in an efficient manner.

1.3.1.2 Improved variants of PRMA

Several modified and improved variants of PRMA have been proposed (e.g., DPRMA, C-PRMA, DRMA), aimed at providing better service for VBR traffic. All these protocols have the same frame structure as PRMA. In DPRMA, all users in the system, including data sources, are allowed to reserve bandwidth. The base station is responsible for dividing the bandwidth among the active sources based on their requirements. A mobile station conveys its bandwidth requirements (quantized into a number of discrete levels) to the base station via a few reservation request bits included in the header of every uplink slot. Time critical traffic can preempt the requests made by data sources. DPRMA makes it possible for a VBR source to reserve bandwidth for a sequence of packets to be sent at some requested rate, without forcing it to contend for every single slot along the way. Although the overhead incurred by the wasteful full-slot contention is somewhat reduced by this approach, it is not completely eliminated. Moreover, the chunks of bandwidth allocation are sized as consecutive powers of two, which may cause fragmentation and become another source of significant wastage.

With C-PRMA, all available slots comprise the logical *signaling channel*, whereas all reserved slots constitute the *information channel*. To reduce the cost of contention, slots in the signaling channel are subdivided into contention minislots, which are used by the mobiles to send their reservation requests to the base station. The partitioning between the two logical channels is managed by the scheduling algorithm run by the base station. Instead of signaling the slot status at the end of every slot, the base station explicitly polls the mobile stations at the moments when they are allowed to transmit. The protocol is aimed at microcellular environments, where the cost of polling is low and where using minislots (rather than packet slots) for contention results in a significant reduction of collision overhead. The information transmitted in a contention slot by a mobile station includes the amount of time spent by its topmost packet in the output buffer. The base station notifies those mobile stations that have succeeded in the last contention via a special message

(command), so that they can refrain from posing further requests on the signaling channel. The scheduling algorithm will decide when the base station should poll those mobiles for their packets, which need not happen in the very next frame.

1.3.1.3 D-TDMA and RAMA

D-TDMA was proposed as a protocol suitable for a variety of multiple access scenarios, including satellite channels [21] as well as PCS environments [22]. Time on the channel is divided into a contiguous sequence of TDMA frames, which are further subdivided into request slots, voice slots, and data slots, as shown in Figure 1.2(b). The basic channel access scheme follows the same idea as in PRMA, which consists of allocating a fixed periodically reserved portion of the frame to voice calls, and making whatever is left available to non-periodic, low priority data traffic. Requests for bandwidth are issued by the mobiles using slotted ALOHA within randomly selected request slots, which are shorter than the slots used for packet transmission. A station generating a new voice talk-spurt or a new data packet randomly picks one of the reservation slots in the next frame and transmits a reservation request packet. If the reservation packet collides with others, the station will not receive an acknowledgment, which would otherwise arrive on the downlink channel before the end of the current frame. An unacknowledged (i.e., timed out) voice request will be reissued by the mobile in the very next frame, while a failed data request will be rescheduled with a randomized delay. A voice source receiving an acknowledgment of its request is assigned one of the available voice slots, which it will keep in subsequent frames until the end of the talk-spurt. Successful requests for data transmission are queued by the base station and serviced in the FCFS manner. The base station notifies the queued data sources when they can commence their transmissions.

A single data packet is allowed to span multiple consecutive slots. The base station tries to allocate voice slots from the beginning of the frame and keep the allocated voice area contiguous. Any unallocated voice slots following the last allocated voice slot can be used by data packets. Additionally, some slots at the end of the frame are reserved specifically for data, and they can never be used for sending voice.

The only difference between D-TDMA and RAMA is in the channel access strategy. The frame structure of RAMA is identical to that of D-TDMA, except that the reservation slots are renamed *auction* slots. Instead of slotted ALOHA, RAMA uses an *auction strategy* to achieve a higher probability of success. A mobile station contending for bandwidth starts by generating a random number called the station's ID. The number of digits in the ID

depends on the number of users in the system: it must be big enough so that two mobiles are unlikely to generate the same ID and small enough to make the auction procedure short. At the beginning of the auction, the contending mobile will transmit its ID, one digit at a time. Following each transmitted digit, the base station will announce the highest value among the received digits on the downlink channel. Any source with the current digit value less than the announced one will drop out. When all the digits have been transmitted, there will be a single winner. The stations dropping out from the current auction enter a new auction at the end of the current one.

1.3.1.4 DQRUMA

The frame structure of DQRUMA is shown in Figure 1.2. Similar to C-PRMA, the uplink channel is logically divided into two subchannels, dubbed the *request-access* (RA) channel and the *packet transmission* (Xmt) channel. This division is accomplished on a slot-by-slot basis. In particular, the base station may convert idle packet slots (i.e., the idle transmission channel portions of the frame) into additional request slots, as needed. Whenever a mobile station wishes to transmit a packet, it contends on the RA channel to convey its request to the base station. The access protocol can be either ALOHA or a deterministic tree-based collision resolution protocol, similar to the one originally proposed in [26]. When the base station successfully receives a request, it transmits an ACK on the downlink channel and registers the request in a queue. The base station then arbitrates access to the uplink Xmt channel in a round-robin fashion by polling the mobile stations on the downlink channel. When a mobile is polled, it transmits the next outgoing packet appending to it one extra bit indicating whether the station has more packets to transmit. By piggybacking its requests onto transmitted packets, the mobile does not have to contend for channel access for as long as its packet queue is nonempty.

1.3.1.5 Summary

All the protocols mentioned above have a similar channel structure. Time on the uplink channel is divided into frames, and each frame is divided into a number of equal length transmission slots and a number of possibly smaller minislots used for contention resolution. This approach is efficient if all mobiles have the same traffic patterns and bandwidth requirements, e.g., as in a cellular voice system. In a scenario with diverse applications involving VBR and data traffic, some protocols (e.g., D-TDMA and RAMA) have the flexibility to assign two or more transmission slots to one mobile station, as needed to satisfy its

dynamic bandwidth requirements. One problem with this solution is that the granularity of bandwidth assignment is constrained by the slot size or, as in DPRMA, by some multiples of slot size. Since the slot length is usually tailored to efficiently accommodate one traffic type, i.e., voice, it may not fit very well the requirements of non-CBR traffic.

Even the flexible schemes that can allocate multiple slots to a single source suffer from a bandwidth wastage resulting from the fact that each of the multiple slots requires individual "framing." This framing consists of a guard time at the beginning and end of the slot equal to the maximum propagation delay across the cell, as well as a synchronizing preamble (see Figure 1.2(b)) preceding the actual data. Therefore, if multiple slots are assigned to the same mobile in one frame, some bandwidth is wasted on multiple slot boundaries within a *de facto* single transmission unit (clearly visible if the multiple slots are adjacent, e.g., as in D-TDMA). This overhead tends to increase linearly with the transmission rate and cell size.

More recent variations on the above schemes, e.g., [27, 28, 29, 30], follow essentially the same paradigm, focusing on bandwidth scheduling and analytical performance studies. They all make bandwidth allocation based on fixed-size slots, usually corresponding to ATM cells. In [27], mini-slots are used for making access requests; normal fixed-size slots are used for transporting data traffic; plus, a centralized scheduler is designed for CBR, VBR and ABR at the base station. [28] has the similar structure as [27] while the priority scheme is considered in [28] and different traffic models are used for protocol performance evaluation. In [29], the time frame is divided into several sections; each section is assigned to one type of traffic, e.g., CBR, VBR, SCR (signal control), and ABR. CBR and VBR are managed with reservation model, SCR and ABR work with contention mode. The scheduler at the base station consists of multiple access controller, traffic estimator/predictor and intelligent bandwidth allocator. [30] mainly deals with how to save bandwidth on piggyback requests. A quantitative comparison of seven TDMA-based MAC protocols is given in [31]

1.3.2 CDMA-based MAC Protocols

CDMA is viewed today as the most promising technology for developing flexible, high-performance wireless systems. Wideband CDMA has been selected as the basis for third-generation (3G) mobile systems [6]. A 3G communication system will offer session bit rates between 384kb/s and 2Mb/s. To support services within this range the radio network resources must be shared in a flexible and efficient manner, such that applications with different bit rates and signal quality (expressed as the minimum bit energy-to-noise density

ratio E_b/E_0) can comfortably coexist in the system, making the best possible use of its total capacity. To achieve this, various CDMA-based medium access control (MAC) protocols have been proposed, e.g., Slotted ALOHA DS-CDMA [32, 33, 34, 35], VSG-CDMA [37], MC-CDMA [36], WISPER [40], and MC-CDMA/DQRUMA [41, 42].

1.3.2.1 Slotted ALOHA DS-CDMA

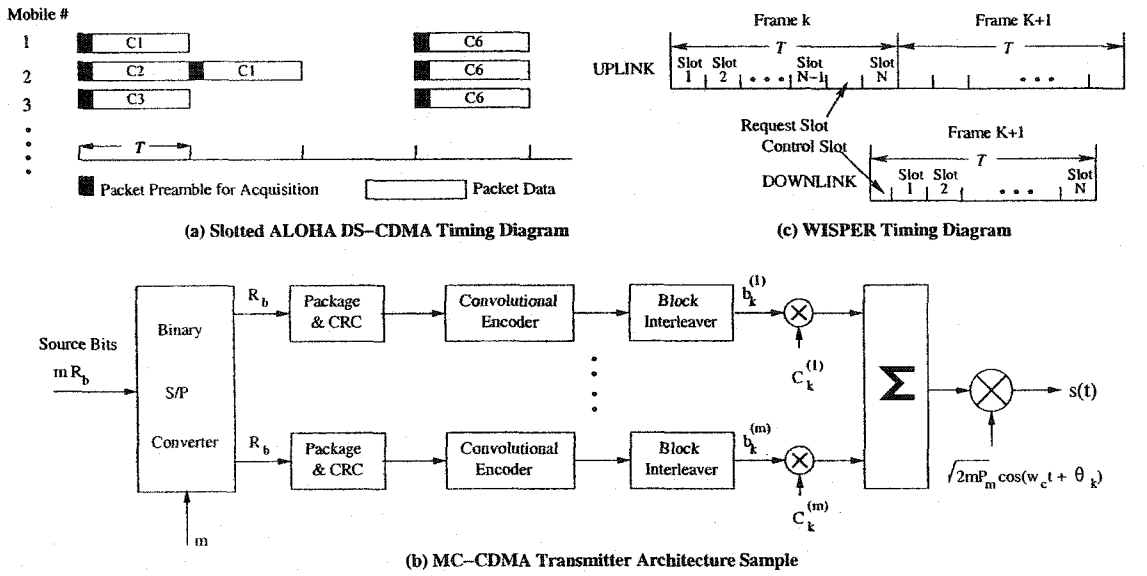


Figure 1.3: Layouts of typical CDMA-based protocols

Slotted ALOHA DS-CDMA is one of the most studied CDMA protocols. In a Slotted ALOHA DS-CDMA system, the mobile stations communicate with the base station through a common radio spectrum. Time is divided into slots of duration T which equals the time to transmit one data packet, as shown in Figure 1.3(a). The mobile users are synchronized to slot boundaries and their transmissions are only allowed to start at known predictable instances. Transmissions from different stations are spread with different codes. A dedicated receiver for every code being currently in use is assigned at the base station.

The code allocated to a mobile user can be fixed [32, 34] or randomly selected by the mobile station [33]. When a mobile has a packet to send, it spreads the packet with its code and transmits the encoded signal in the next time slot. Depending on the channel status and multiple access interference (MAI) level, some packets may be undecodable at the base. This will happen, when two mobiles use the same code for transmission or when the MAI level exceeds a certain threshold. The latter problem is caused by too many mobiles transmitting at the same time, or by misadjusted power levels used by the transmitters.

Finally, owing to the inherent unpredictability of the radio environment, packets can be damaged by external sources of electromagnetic interference and/or by fading.

If a packet cannot be received correctly for whatever reason, the sender will not receive an acknowledgment and then it will schedule a retransmission of the same packet after a random delay. Under light load, the slotted ALOHA DS-CDMA protocol works well [35], but under heavy load, numerous collisions tend to dramatically reduce its effective bandwidth utilization. Also, Slotted ALOHA DS-CDMA does not differentiate among traffic classes, which is essential in modern multimedia applications.

1.3.2.2 Variable spreading gain CDMA (VSG-CDMA)

Variable spreading gain CDMA (VSG-CDMA) [37] is being frequently suggested as the right technology for MAC protocols catering to variable data rate multimedia applications. With this approach, different transmission rates are realized by using different spreading factors. If two baseband signals at rates r_1 and $r_2 = 2r_1$ are spread onto the same bandwidth, the signal with rate r_2 uses half the spreading factor of the signal with rate r_1. In a VSG-CDMA system, high speed data transmission requires more power to satisfy the same signal-to-interference (SIR) ratio as one of a lower data rate.

Based on the VSG-CDMA technology, various MAC protocols have been discussed in the literature [37, 38, 39]. With the adaptive multiple access protocol analyzed in [37], the base station estimates the equivalent uplink load by monitoring the total received power, and broadcasts this information to the mobiles, which then make independent probabilistic decisions on whether to begin and/or to continue transmitting their bursts.

Another MAC protocol, intended for UMTS terrestrial radio access system (UTRA), is presented in [38]. It supports constant bit rate (voice), variable bit rate (video), and low priority data services. According to this protocol, mobile stations make requests for dedicated channels (DCHs) in time-slotted random access channels (RACHs), as shown in Figure 1.4. If there has been no collision (caused by more than one access request transmitted within the same slot and with the same code), the mobile will receive a positive acknowledgment from the base station on the forward access channel (FACH), followed by further instructions. If the access request collides, the mobile will delay for a randomly chosen number of slots before posing another request. The base station stores all incoming requests in three FIFO queues, one for each traffic class, and assigns the DCHs according a code tree, as shown in Figure 1.5. A code can be assigned to a mobile only if all associated codes at lower levels of the tree are idle. Each DCH is identified by a spreading code whose

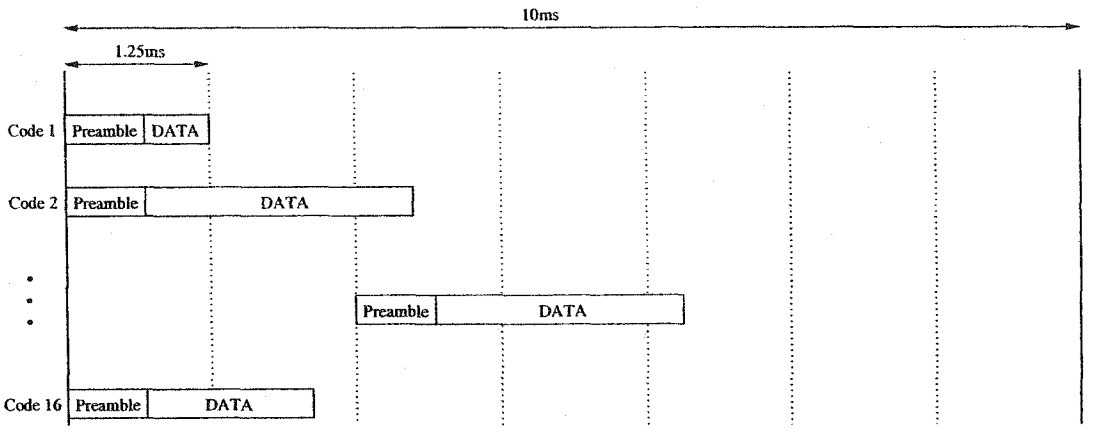


Figure 1.4: RACH structure in UTRA

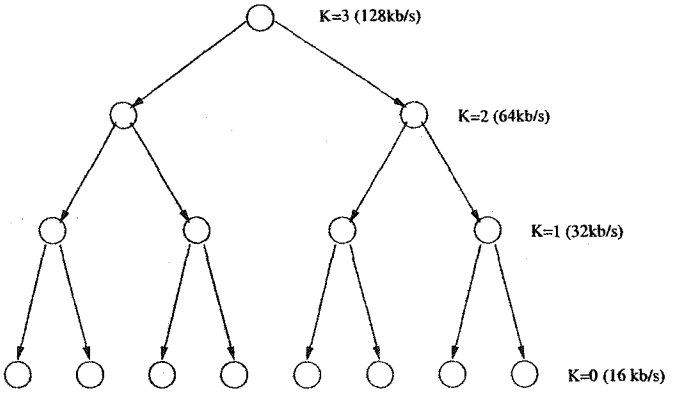


Figure 1.5: A sample code tree

spreading factor determines the bit rate of the channel. Since this protocol assigns DCHs based on the number of idle codes, and these assigned codes remain reserved for the entire duration of a CBR session, this protocol cannot reuse silent periods in voice traffic.

The protocol proposed in [39] supports two classes of services: class-1 (real-time) and class-2 (reliable). The protocol is similar to slotted ALOHA, but the class-2 mobiles are subject to media access control with their spreading gain adapted to the multiple access interference (MAI) level. This adaptation consists in decreasing/increasing the mobile's spreading gain in response to the decrease/increase in the MAI level. Also, the persistence factor for class-2, i.e., the retransmission probability after an error, is always 1. Note that a packet can be retransmitted with a spreading gain different from the one used for its original (previous) transmission. The objective of this approach is to reduce the number of retransmissions within class-2 and to maximize its effective throughput.

In the protocol mentioned above, the spreading gain changes with the symbol duration while the chip rate is kept constant. An alternative approach is to vary the chip rate while maintaining a constant symbol duration. Regardless of which approach is taken, the spreading gain may turn out to be too small to to maintain a good (low) cross correlation among different user codes, especially when the source rate is very high. Also, the requirement to frequently change the chip rate or the symbol rate may increase the hardware complexity of the code acquisition/tracking circuits (or the symbol demodulation circuits) beyond an acceptable level.

1.3.2.3 Multi-code CDMA (MC-CDMA)

Multi-code CDMA (MC-CDMA) [36] is another heavily studied CDMA technology for implementing flexible multiple access schemes to support high data rates and multimedia applications in wireless communication systems. In a multi-code CDMA system, all transmissions are carried out at a fixed basic rate. If a high data rate is required, multiple code channels are assigned to the mobile, with each single code channel operating at the basic rate. Thus, MC-CDMA is a fixed spreading gain CDMA technology, which lets it avoid some problems with VSG-CDMA. Figure 1.3(b) shows a sample structure of an MC-CDMA transmitter. When a mobile transmits at a rate equal to m times the basic rate, it converts its data stream, serial-to-parallel, into m basic rate streams, spreads each basic rate stream with a different code, and superimposes them before up-converting for radio transmission. As in VSG-CDMA systems, high speed data transmission in MC-CDMA systems requires more power to satisfy the same signal-to-interference (SIR) ratio as that of lower data rate.

A number of MAC protocols have been proposed based on the generic MC-CDMA approach sketched above [40, 41, 42, 43]. In particular, the slotted CDMA protocol introduced in [40], called *WISPER*, is able to support multiple traffic classes. The total available bandwidth is divided into two bands, one for the uplink, the other for the downlink. For both bands, time is divided into frames of length T. The frame length is chosen so as to coincide with the packet arrival rate of the most abundant traffic class (usually voice). In the uplink, each frame is divided into N packet slots and one request slot. The structure of the downlink channel is similar, except that the downlink frame begins with a control slot, which is used to transmit acknowledgment and control commands, whereas the request slot in the uplink frame can be located anywhere except for the very end of the frame. Also, the downlink frame is not aligned with the uplink frame. The layout of the frame in WISPER is shown in Figure 1.3(c).

The transmission order of outgoing packets is determined by the packet scheduler, which bases its decisions on the priority and QoS requirements of the sessions in progress. The latter are described by the admissible bit error rate (BER) and time-out vales, i.e., deadlines. The scheduler also keeps track of the number of queued packets at each mobile and attempts to service the multiple queues in a fair manner. Packets with the same or similar bit error rate (BER) requirements are transmitted in the same slots. One problem with WISPER is that the strict policy of assigning similar packets to the same time slots (which balances the load along the code dimension) may result in an unbalaced load along the time dimension.

The protocol proposed in [41] combines MC-CDMA with DQRUMA [24]. As in [24], time is divided into slots. The downlink frame is partitioned into three types of time slots: access acknowledgment slots, packet acknowledgment and transmit permission slots, and packet transmission slots. The uplink frame consists of two slot types: access request slots and packet transmission slots. Within each time slot, multiple codes separate different logical channels. Mobile stations issue access requests to the base station in the request channels, and base station schedules the requests using a bandwidth-on-demand fair-sharing round-robin strategy. The actual number of packets that a mobile is allowed to transmit depends on the current load (the number of outstanding requests), the transmission rate of the mobile, and the available system capacity.

The base station also calculates the optimum power level for each traffic rate. Although this protocol provides variable transmission rates, it only considers one value of the signal-to-interference ratio (SIR) and, consequently, supports only one traffic class. With the improvement suggested in [42], the protocol can cater to two traffic classes. It uses a similar frame structure to that of [41]. The uplink time axis is divided into frames, with each frame consisting of three short control slots (minislots) and one packet transmission lot. The three minislots are used for acknowledgments, access contention, and piggybacking rate adjustment requests for sessions in progress. Similarly, every time frame on the downlink channel includes two minislots in addition to the standard transmission slot. Those minislots are used for sending acknowledgments and for announcing the outcome of access contention. As in [41], multiple logical copies of the control minislots and transmission slots are created by using different PN codes.

Class-I (real-time) traffic is handled through connection-oriented sessions, which reserve a set of code channels and dedicated receivers at every participating mobile. Class-II (non-real-time) traffic is transmitted in a best-effort manner through a contention based request-permission scheme. One problem with this protocol is its inability to reuse the bandwidth

released by temporarily silent voice sessions.

1.3.2.4 Summary

The capacity of CDMA systems based on pseudo-noise (PN) sequence spreading is interference limited. Their bandwidth efficiency can be improved by exploiting inactive periods in voice sessions. Most MAC protocols proposed in the literature accomplish this through monitoring the level of multiple access interference. For example, the protocols proposed in [43, 44] grant transmission rights based directly on the interference level, with [43] focusing on the management of high rate traffic. With the probabilistic access control schemes presented in [45, 46], the permission probability for data traffic is affected by voice activity. Although such schemes implicitly exploit idle periods in voice sessions, they base future scheduling decisions on extrapolations of past activities. Consequently, the system may occasionally become underloaded or overloaded, if the predictions turn out to be inaccurate.

One fundamental problem of a wireless system catering to multiple traffic classes with different transmission rates is fairness, especially the treatment of high-priority low-rate terminals under a dominant high-bandwidth load. Some authors, [43, 48, 47], suggest to assign bandwidth to high-rate users based on their burst level demands. The solution discussed in [43, 47] grants access to a high-rate source by considering channel load, interference, and soft handoff. To protect voice users, a high-rate data burst can be preempted. In [48], two transmission modes are considered for high-rate sources, and it is demonstrated that by carefully scheduling those transmissions, their throughput can be maximized without impairing voice sessions. All those studies assume the existence of a low-rate dedicated signaling channel.

Chapter 2

DS-TDMA/CP

TDMA is one of the generic wireless multiple access technologies studied in our work. With TDMA, the uplink channel is time-divided among the multiple mobile stations that may try to reach the base station at the same time. The mobile stations learn which time intervals (dubbed transmission slots) are theirs by receiving and interpreting control information arriving from the base station on the downlink channel. The uplink channel is logically organized into frames, each frame subdivided into a number of slots, which are the basic and indivisible units of bandwidth allocation. The role of the frame is to provide encapsulation for groups of slots (a "frame of reference") whereby all mobile stations can have a consistent perception of slot boundaries and their positions relative to the frame boundaries. Depending on the details of the access protocol, the frame may include additional components (e.g., synchronizing signals, acknowledgments, contention slots), which are emitted by the base station on the downlink channel.

2.1 The protocol goals

Our proposed TDMA protocol is dubbed DS-TDMA/CP, for *Dynamically Slotted TDMA with Contention Permission.* It is aimed at improving the accuracy of bandwidth allocation, especially in environments with diverse traffic patterns and QoS expectations.

Whereas other TDMA-based schemes proposed in the literature use fixed-size slots within a frame and try to fit the offered traffic to that size, our solution uses an adjustable slot size that is matched to the bandwidth requirements of the mobile source, i.e., the frame structure is determined by the transmission requirements of the mobile stations. The idea is to make sure that the mobile will always use at most one slot of every frame, yet its bandwidth requirements will be satisfied and the QoS requirements of its session will be fulfilled. In this way we eliminate bandwidth fragmentation and avoid the overhead on

multiple slot boundaries. This solution offers better flexibility of bandwidth allocation than traditional methods, which directly translates into a lower bandwidth overhead and better fulfillment of the QoS requirements of the diverse traffic types occurring in modern PCS applications.

We assume that traffic in the network is naturally packetized with some basic granularity that is also used for bandwidth allocation. For the sake of discussion, we assume that this granularity is constrained by ATM cells, i.e., the network carries ATM traffic. However, this is not a requirement, and our proposed solution can be naturally applied to non-ATM networks. Also, the bandwidth allocation grain can be arbitrarily fine (e.g., much finer than ATM cells) with little impact on the complexity or flexibility of the overall scheme.

2.2 Silent periods in CBR traffic

Voice traffic, which is the most common representative of the CBR class, is known to be intermittent with a duty factor of approximately 3/8 [70]. Some TDMA protocols assume that individual bursts (talkspurts) in a CBR session are separate "sessions" of their own, which have to individually request bandwidth as they occur. With this approach, it may be possible to transmit non-CBR traffic during silent periods of a CBR session [71]. One problem with such protocols is that they admit the possibility of a collision when the voice source resumes its talkspurt, and the station may lose its reservation in such a case.

An alternative approach is to keep the CBR slot reserved for the entire duration of the actual voice session, including the silent periods, just to make sure that it is there when needed to accommodate a burst. Of course, this approach is more wasteful because the bandwidth used to sustain a silent CBR session, although formally not needed, cannot be used for anything else. Unless the protocol explicitly accounts for the presence of CBR sessions that have temporarily become silent (and treats them in a special way), the choice between the two options need not belong to the protocol specification.

To be able to reuse the bandwidth temporarily relaxed by a silent CBR session, in a way that will make that bandwidth available at the next burst, the base station must receive advance notifications about the status of the current burst (to learn when it is going to end), and be able to accommodate the new burst, preferably without excessive and nondeterministic contention, when it shows up. The concept of piggybacking seems to be an essential prerequisite for this capability. For example, if the only way for the base station to learn that the current burst has ended is to receive an empty CBR slot (like in PRMA or D-TDMA), the empty slot is irreparably wasted. On the other hand, in DQRUMA, as

stations piggyback their advance bandwidth requirements onto transmitted (full) slots, a CBR source can indicate that the burst is about to end early enough for its next slot to be reassigned to another mobile.

This feature must be combined with a way of reverting a silent CBR session to its active state at the beginning of the next burst. Although the bandwidth scheduler at the base station can freely preempt less important sessions to make room for the new burst, it must be able to learn that the burst is there, i.e., that the silent session has become active. Ideally, the mobile should keep a tiny portion of its original bandwidth (slot) while in the silent state—just enough to be able to "piggyback" the new status of its session. With the rigid organization of the frame (where bandwidth comes in fixed-size slots) this approach is not feasible. A compromise solution may be to give a resumed CBR session a head start when competing for access to the base. In particular, the deterministic variant of the contention scheme in DQRUMA may provide an acceptable bound on the amount of time needed by a CBR source to notify the base about the status change of its session.

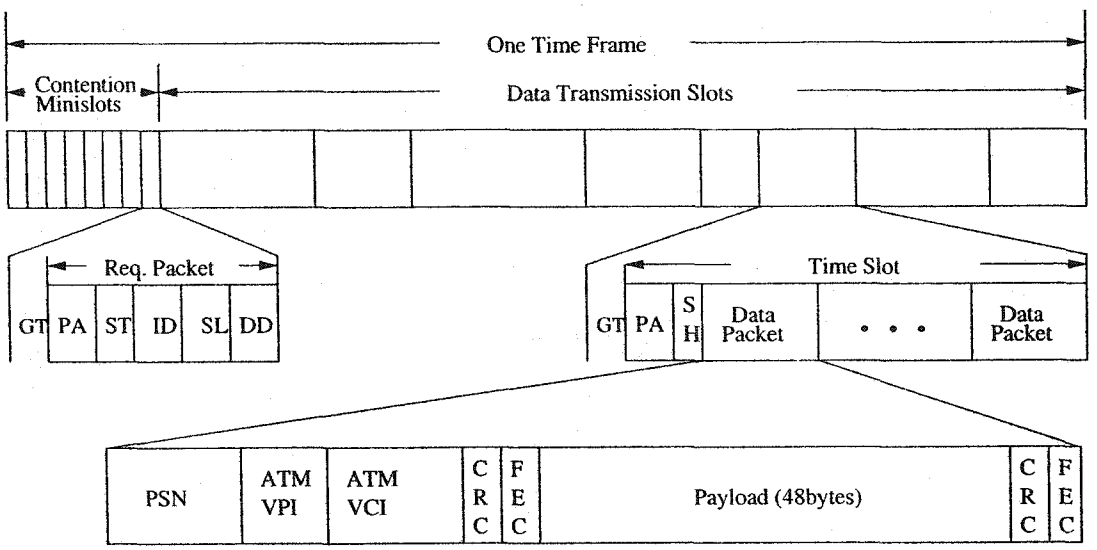
2.3 DS-TDMA/CP Protocol Description

As indicated in Section A.5, DS-TDMA/CP is designed for cellular radio communications system. We assume that the downlink channel is separate from the uplink channel in radio frequency bands and that its straighforward implementation is uninteresting from the viewpoint of the access scheme needed for the uplink channel.

2.3.1 Frame structure

Assume that the traffic in the mobile network consists of ATM cells, i.e., the base station is connected to an ATM network, and the mobile stations access the ATM network through the base. Time on each of the two channels (downlink and uplink) is divided into equal length frames. The frame length is set to coincide with the packet arrival rate of the CBR traffic.

The only station allowed to transmit on the downlink channel is the base and, therefore, the management of this channel is trivial. Consequently, we shall consider the uplink channel only. The uplink frame structure used by DS-TDMA/CP is shown in Figure 2.1. The frame is divided into two sections. The first (contention) section consists of a sequence of minislots, much shorter than a typical transmission slot, used by the mobile stations to issue access requests. The second (transmission) section is dynamically partitioned into a number of variable-length transmission slots, according to the current bandwidth assignment to the



GT: Guard Time
PA: modem Preamble
ST: Service Type (CBR/rt–VBR/nrt–VBR/ABR/UBR)
ID: User Identification Number
SL: Time Slot Length
DD: Due Date

SH: Slot Header
PSN: Packet Sequence Number
ATM VPI: ATM Virtual Path Identifier
ATM VCI: ATM Virtual Channel Identifier
CRC: Cyclic Redundancy Code
FEC: Forward Error Correction

Figure 2.1: Uplink frame structure in DS-TDMA/CP

mobile stations.

2.3.2 Principles of bandwidth allocation

Consider the following traffic classes listed in the decreasing order of their priorities:

CBR	bursty traffic with constant bit rate and tight deadlines
RT-VBR	variable bit rate traffic with deadlines that are less tight than those of the CBR class
NRT-VBR	variable bit rate traffic with loose deadlines
ABR	higher priority traffic that can be delivered according to a reasonable best effort policy
UBR	low priority traffic to be delivered according to the best effort policy

An access request packet sent in a contention minislot includes the traffic class (i.e., expected service type), the mobile ID, the requested length of the transmission slot, and the deadline (*due date*), after which the request should be dropped if it cannot be granted. Note that a CBR source need not specify its bandwidth requirements because they are assumed to be

the same for all CBR sessions. The base station stores the incoming requests and grants them according to the criteria explained below. The transmission section of the frame is used for transmitting packets. The size of the variable length slot allocated to a mobile in the transmission section mainly depends on the mobile's transmission rate. The station may keep its slot reserved in subsequent frames until it runs out of packets or is preempted by the base station. A CBR source is guaranteed to receive its slot in every frame until the end of its session, i.e., CBR sessions are never preempted.

Slots are allocated with a granularity that lets a mobile station transmit an entire number of packets within a slot. To be compatible with ATM, a single packet corresponds to one ATM cell. Its format is shown in Figure 2.1. Each transmission slot begins with a guard time and a synchronizing preamble. The end of the preamble indicates the moment when the mobile to which the slot has been assigned may commence its transmission.

The sequence of packets transmitted within the slot is preceded by a header in which the mobile piggybacks its further requirements for bandwidth. In particular, if the station has been transmitting CBR packets and its burst has run out, it will indicate in the slot header that its slot should be released in the next frame. Similarly, a station sending VBR traffic will indicate in the slot header the requested size of the next slot and the due date for this request.

2.3.3 Bandwidth scheduling

The structure of the next uplink frame is announced by the base station (on the downlink channel) a moment before the frame is started. This announcement consists of the following information:

- the contention permission flags (CPF) indicating which traffic classes are allowed to compete for bandwidth
- the number of minislots in the frame (S)
- the list of scheduling messages consisting of pairs: <*station Id*, *slot length*[1]>

The number of flags (bits) in CPF is equal to the number of traffic classes, e.g., 5 according to the above list. A traffic class is allowed to compete for bandwidth (i.e., transmit request packets within the current frame), only if its permission flag in the frame announcement was set. The role of CPF is to selectively restrain some traffic types when bandwidth becomes scarce. This is the Contention Permission feature of DS-TDMA/CP.

[1]Expressed in ATM frames.

As we shall see in Section 2.4.2, the actual information sent in a frame announcement message is a bit disordered, e.g., the list of scheduling messages precedes the permission flags and the number of minislots. Based on that information, the mobile stations have to perform some simple calculations to determine the actual succession of their slots in the forthcoming frame. This is needed to simplify the operation of the bandwidth scheduler at the base station, so that it can expedite the frame announcement on time—before the frame becomes due.

The boundary between the two sections of a frame, solely determined by the number of minislots S, varies according to the load in the network. When the load is heavy, which means that few (if any) new requests can be accommodated at this time, the base station will issue a frame with a short contention section and a restrictive setting of *CPF*—to accept high-priority requests only. Theoretically, in an extreme case, e.g., if the frame is tightly occupied by CBR sessions that have all become active, the number of minislots may be zero. However, it practically never happens in reality, because even a heavily loaded frame usually contains a small fragmented leftover that is turned into contention minislots (see the scheduling algorithm in Section 2.4.2). For example, in our experimental setup (Section 2.5.2), the minimum number of contention minislots was effectively 6. On the other hand, when the system load is light and there is little to transmit, the base will issue frames with long contention sections—to make the contention easier and quicker to resolve.

The boundary between CBR (highest priority voice traffic) and VBR (second highest priority flexible-bandwidth video) constitutes a special case from the viewpoint of bandwidth availability. Due to the burstiness of voice activity, as discussed in Section 2.2, the *CPF* for the second highest priority traffic class, i.e., VBR, can be set on all the time. In this way, the silent periods in voice sessions can be always reused, regardless of the CBR load level.

Suppose that a previously idle mobile station gets a packet to transmit and, based on the current setting of *CPF*, its traffic class is allowed to compete for bandwidth in the forthcoming frame. The station will randomly select one minislot in the contention section of that frame and transmit a reservation packet. This packet may make it to the base station or be destroyed, e.g., by a collision with another reservation packet sent by some other station. Other (piggybacked) reservation requests may arrive in the headers of regular slots transmitted within the frame.

When the base station has received the header of the last slot of the frame, it will be ready to process new requests and schedule bandwidth for the next frame. The base station will try to accommodate the successfully received reservation requests, based on

the available air-time, the traffic class priority, the requested slot length, and the specified deadlines. Then the station will broadcast the layout of the next frame to the mobile stations.

If the mobile's request has made it to the base station, the mobile will receive a scheduling message, even if its requirement cannot be met at the moment. In such a case, the slot length field of that scheduling message will be set to zero. This will inform the mobile that its request has been noticed and need not be reissued in subsequent frames. The mobile will refrain from further contention for the amount of time equal to the specified time-out (due date) of the request. Similarly, if the base station cannot fulfill this request before the due date, it will drop the timed-out request from its queue. On the other hand, if the request packet has been lost (e.g., due to a collision), the mobile station will receive no scheduling message. Then it will know that the request should be retransmitted as soon as possible.

Note that all scheduling messages should be able to reach the mobile stations before the beginning of the next frame. The amount of time during which the base station has to make the scheduling decisions and transmit the packet announcing the layout of the upcoming frame is bounded by the length of the last slot in the current frame plus the inter-frame space. To maximize this time, the longest slot is always scheduled at the end of the uplink frame.

The protocol can operate correctly in a situation when the last slot in an uplink frame is too short for the base station to complete the scheduling algorithm and/or broadcast the scheduling messages before the next uplink frame is due. With contemporary technology, it seems that a situation like this can be avoided, however, a cheaper design of the base station (with less processing power available to the bandwidth scheduler) may be susceptible to such problems. If a scenario like this should be taken into account, the base station will set up a timer at the beginning of every frame. This timer will go off at the last moment when the base must complete its scheduling decisions and announce the layout of the next uplink frame. The base station has two options regarding the fate of a received but unscheduled request. One option is to ignore such a request which, from the viewpoint of the mobile, has the same implication as a collision. The mobile will assume that the request hasn't made it to the base and reissue it in a subsequent frame. Another option is to acknowledge the request with zero allocated bandwidth. In that case, the mobile will not have to compete again for access to the base, at least for the amount of time equal to the deadline of the request.

Another degree of freedom that the base can exploit if the timing of the scheduling message becomes tight, is the possibility of skipping the part of the announcement message that indicates the number of minislots available in the next uplink frame and the setting of *CPF*. In particular, if these parameters are going to be the same as for the previous uplink frame, the base can simply skip them, and the mobiles will assume the last received setting by default.

By allocating the bandwidth for the next uplink frame at the very end of the previous frame, the base station is able to account for as much air-time in the next frame as possible, which in turn results in high flexibility and promptness in granting requests of the mobile stations. This is because those mobile stations that have been allotted space in the current frame piggyback their new bandwidth requests, including bandwidth relaxation, in their slot headers.

Having received a scheduling message indicating its bandwidth assignment, a mobile station will start transmitting data packets in its allocated slot. If the traffic is non-CBR, the station will monitor its output buffer during transmission and piggyback slot adjustment requests in the header of its slot. If the buffer drains faster than the traffic is generated, the mobile will release some fraction of its slot. If the buffer grows faster than the data that is being expedited, the station will request more bandwidth.

Once a CBR connection has been set up (meaning that the request has been scheduled by the base station), the mobile will keep receiving a slot in every subsequent frame until the connection is explicitly torn down by the mobile. If the connection becomes temporarily silent (see Section 2.2), its bandwidth can be reused by non-CBR sources. Instead of using the headers of its slots to convey bandwidth adjustment requests, a CBR source indicates there the status of its connection for the next frame. This status can have three values: *active*, meaning that the burst (talkspurt) is still on and the next slot will be filled, *hung up*, meaning that the connection is terminated and the slot should be released, and *silent*, indicating that the connection should remain established but the next slot will be empty. In the last case, the base station is free to temporarily reduce the slot size in subsequent frames to "almost zero," i.e., to the bare header. When the connection gets back to the *active* state, the mobile will use that header to indicate the state change to the base station. In this way a silent voice connection remains established by using very little bandwidth. The mobile station will not have to contend for bandwidth when the silent period is over.

2.4 Algorithms in DS-TDMA/CP

In this section we introduce in detail the algorithms used by the protocol. They can be viewed as sample (or recommended) implementations of the ideas introduced in the preceding section. These implementations were also used in our simulation model of DS-TDMA/CP discussed in Section 2.5.2.

2.4.1 Contention resolution

Access request packets transmitted within the contention minislots of a frame may collide, if two or more mobile stations happen to randomly pick the same minislot. As long as the protocol is consistently obeyed by all stations, this is the only scenario in our proposed scheme when a collision may occur.

The contention resolution part of DS-TDMA/CP can be implemented independently of the remaining elements of the protocol. For simplicity, we opt here for a variant of slotted ALOHA inspired by [42, 24]. When a mobile station is ready to issue a request, it randomly selects a contention minislot and transmits the access request packet. If the request makes it to the base station, the mobile will hear a scheduling message with its own ID on the downlink channel. If this doesn't happen within the current frame, the mobile station will assume that a collision has occurred and make another try (in the contention section of the next frame) with the probability

$$P_n = \min\left\{\frac{S_n}{S_c} \times \frac{P_c}{P_c + 1}, 1\right\}$$

where P_c is the retransmission probability used for the previous request (set to 1 for the first attempt), S_n is the total number of contention minislots available in the upcoming frame, and S_c is the number of contention minislots in the last frame. This backoff algorithm resembles the harmonic backoff algorithm (with retransmission probabilities 1, 1/2, 1/3, 1/4, ...) described in [71, 24], and additionally accounts for the dynamic structure of the contention section. Our backoff function reduces to the harmonic backoff, if the length of the contention section remains fixed across frames. However, if the number of contention minislots in the upcoming frame is reduced, the retransmission probability will decrease faster than with the straightforward harmonic algorithm. Similarly, as more contention minislots become available, the retransmission probability will decrease by less or, possibly, even increase.

Needless to say, other collision resolution protocols, e.g., deterministic ones based on the binary tree algorithm [26, 24], may be good alternatives to ALOHA. Although the

performance of DS-TDMA/CP may vary depending on the collision resolution scheme used, the protocol may still be compared with other similar solutions, as long as they all use similar methods of resolving contention.[2] Besides, as other collision resolution schemes are more deterministic (and offer lower delays) than ALOHA, the ALOHA-based version of DS-TDMA/CP can be viewed as the worst case.

2.4.2 Transmission scheduling

We consider the traffic classes listed in Table 2.1. Each of those classes has its specific QoS expectations; therefore, the access requests arriving at the base station are stored in different queues directly corresponding to the traffic classes. Requests in each queue are stored in the non-decreasing order of their deadlines. Requests with the same deadlines (e.g., UBR requests with infinite deadlines) are stored in the order of their arrivals.

Priority	Traffic Class
0	CBR
1	RT-VBR
2	NRT-VBR
3	ABR
4	UBR

Table 2.1: Traffic priorities

The scheduler tests the access requirements of the pending requests to see which of them can be accommodated and moved to the queue of scheduled requests. While doing this, the scheduler generates scheduling messages, one message per each request that has been granted. A non-CBR source that cannot finish its transmission in the current frame (which fact is signaled by a piggybacked request in the header of the currently transmitted slot) is put back into the corresponding access request queue. On the other hand, a CBR source, once it is admitted by the scheduler, will be receiving a guaranteed slot in subsequent frames until the end of its session (during the silent periods, this slot will shrink to the bare header). The scheduler is work-conserving, which means that a packet in a mobile station becomes eligible for transmission immediately upon generation.

The amount of time available to the scheduler to complete its task and broadcast the layout of the new frame is limited by the size of the last slot in the current frame. To make good use of this limited time, the scheduler generates the layout packet on-the-fly, while it is performing bandwidth allocation. Whenever a slot space is assigned to a mobile, the

[2]The implementations of D-TDMA and DQRUMA studied in Section 2.6 both use ALOHA for contention resolution.

scheduler emits a scheduling message[3] (Figure 2.2) identifying the mobile and specifying the size of the allocated slot as a number of ATM frames. The allocation loop of the scheduler can be exited in one of two ways. First, it may happen that there is nothing more to do, i.e., all the available bandwidth has been allocated or there are no more pending requests. Second, an internal scheduling timer may go off indicating that the scheduler must finish immediately because the frame is due. Intentionally, the second case should occur very seldom and possibly never, depending on the processing power available at the base station. We admit it because the protocol can operate sensibly in such circumstances, which means that lower-cost hardware can be used if needed, e.g., because of budgetary constraints. The worst that can happen is that occasionally some (low-priority) requests may not be fulfilled, even if enough bandwidth remains available to accommodate them.

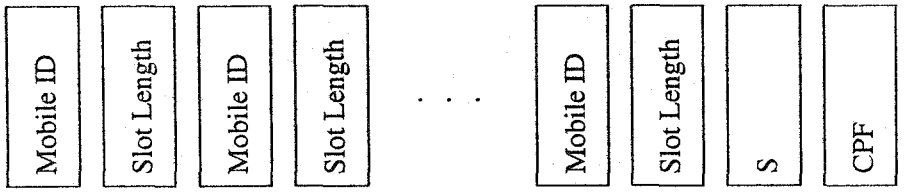


Figure 2.2: Structure of the frame announcement message

Since CBR requests have the highest priority, the scheduler processes them first—in non-decreasing order of their due dates, according to the STE (shortest time to extinction) policy [72]. When all CBR requests have been processed and there is still time and bandwidth left in the frame, the scheduler will process RT-VBR requests (second highest priority), and so on.

Having completed the allocation loop (and generated all scheduling messages), the scheduler emits two more items of information (Figure 2.2): the number of contention minislots in the upcoming frame (S) and the contention permission flags (CPF).

The protocol specifies a maximum on the number of contention minislots in a frame. Note that even if the network is completely idle, the frame cannot be entirely filled with minislots because the last few of them would be too close to the frame boundary to be usable by the scheduler. The actual number of contention minislots (which is never greater than the maximum) and the contents of CPF are determined by how the frame space has been partitioned among the multiple traffic classes and how much bandwidth (if any) remains unused. To understand the rationale behind this part of the scheduling algorithm, consider

[3]The parameters of our experimental setup, discussed in Section 2.5.2, constrain the length of this message to be 20 bits.

the following simple algorithm for determining the number of contention minislots in an upcoming frame.

> We start with a small number of minislots (possibly zero) and allocate the remaining space in the frame to the queued traffic sources, according to the scheduling algorithm sketched above. If some space remains unused (either because there is no more traffic to schedule, or there is a fragmented leftover that cannot be sensibly allocated anywhere), we populate that space with extra minislots, up to the maximum.

Although this simple algorithm has the desirable property that the number of minislots is a straightforward function of the available bandwidth, it fails to account for differences among traffic classes. Imagine two equally heavily loaded frames, one filled with UBR cells and the other carrying exclusively CBR traffic. Even though the two frames may look identical from the viewpoint of bandwidth utilization, it is clear that the numbers of minislots, as well as the settings of *CPF*, in them should be different. The second frame is filled with a high priority traffic, so it makes sense to restrain further contention with a restrictive setting of *CPF* and a reduced number of minislots. On the other hand, the low priority traffic in the first frame is preemptible by all other traffic classes. Those higher priority classes should be allowed to contend for bandwidth, for which they need more minislots and a permissive setting of *CPF*.

It might seem that while deciding on the number of minislots and the setting of *CPF*, the bandwidth scheduler should also consider the state of request queues at the base station. Note, however, that the relevant information is implicit in the frame. For example, if the lowest priority request that has been granted is NRT-VBR and the frame is full, this means that there is no room for ABR and UBR requests, but all higher priority sources should be allowed to contend.

2.4.2.1 The scheduling algorithm

We assume the following notation for the constants and variables used by the algorithm:

Constants (protocol parameters):

B	The total amount of bandwidth (frame space) available in an empty frame.
B_u	The amount of bandwidth (frame space) needed to sustain a silent CBR session. This bandwidth corresponds to the bare CBR slot header.

B_s The amount of bandwidth (frame space) used up by one contention minislot.

C_{max} The maximum index of a traffic class, i.e., 4 according to Table 2.1. Note that there is nothing special about this number, and it may change, if the configuration of traffic classes serviced by the network is different.

S_{max} The maximum number of minislots in a frame.

B_v This is an array indexed by traffic classes indicating the average amount of bandwidth needed by a session in the given class (see the explanation at the end of the algorithm). $B_v[0]$ is exact and equal to the fixed amount of bandwidth needed by an active CBR session.[4]

S_{incr} This is an array indexed by traffic classes used to determine the number of extra minislots to be included in the contention section of the next frame, as explained at the end of the algorithm. The entries corresponding to the first two traffic classes, i.e., $S_{incr}[0]$ and $S_{incr}[1]$, are not used.

Important variables:

S The calculated number of minislots in the upcoming frame.

CPF This is a bit array indexed by traffic classes and representing the calculated setting of the contention permission flags.

B_q The amount of bandwidth temporarily released by those admitted CBR sessions that are currently silent.

B_a The calculated amount of bandwidth still available within the frame, including the bandwidth temporarily released by the silent CBR sessions.

B_{asg} This is an array indexed by traffic classes and indicating how much bandwidth has been allocated to the given traffic class.

The algorithm

1. Set $S := 0$, $B_a := B$, $B_q := 0$, $B_{asg}[0, \ldots, C_{max}] := 0$, $CPF[0, \ldots, C_{max}] := 1$.

2. For all CBR sessions in progress, and then for all pending CBR requests (sorted in the non-decreasing order of their due dates), perform steps 3 through 6, with M_r indicating the mobile source.

[4]Although the present algorithm assumes that every active CBR session uses the same fixed amount of bandwidth, this assumption can be easily relaxed.

3. If there are no more CBR requests to process, proceed to 7. If $B - B_{asg}[0] \geq B_v[0]$ (more CBR sessions can be accommodated), proceed to 4. Otherwise (this is only possible for a pending request), if the request arrived in the last frame, issue the scheduling message $< M_r, 0 >$. Continue at 3 for the next CBR request.

4. If the session is currently silent (this is only possible for a session in progress), set $B_q := B_q + (B_v[0] - B_u)$, $B_r := B_u$. Otherwise (including the case of a pending CBR request being admitted), set $B_r := B_v[0]$.

5. Set $B_a := B_a - B_r$ and $B_{asg}[0] := B_{asg}[0] + B_v[0]$.

6. Transmit the scheduling message $< M_r, B_r >$ and continue at 3 for the next CBR session or request.

7. If $B - B_{asg}[0] < B_v[0]$, i.e., no more CBR sessions can be accommodated (this is only possible after all sessions in progress have been accounted for), set $CPF[0] = 0$. Otherwise, set $S_r = \lfloor (B - B_{asg}[0])/B_v[0] \rfloor$ (the number of extra minislots for the CBR sessions that could be accommodated in the frame), $B_a := B_a - S_r \times B_s$ (extract the bandwidth needed for the additional minislots), $S := S + S_r$.

8. Process the remaining traffic queues according to their priorities, and each queue in the non-decreasing order of the due dates. For each request, perform steps 9 through 13, with

   M_r indicating the the mobile source that issued the request

   C_r indicating the traffic class

   B_r indicating the requested bandwidth

9. If the internal timer went off, which means that there is no time for more scheduling before the frame is due, exit the allocation loop and continue at 14.

10. If there are more requests in the current class C_r, proceed to 11. Otherwise, if $B_a - B_{asg}[C_r] < B_v[C_r]$, set $CPF[C_r] = 0$. If $C_r = C_{max}$, proceed to 14. Otherwise, continue at 9 with $C_r := C_r + 1$.

11. If $C_r > 1$, calculate the number of additional contention minislots to be included in the frame: $S_r = \lfloor (B_{asg}[C_r] + B_r)/S_{incr}[C_r] \rfloor - \lfloor B_{asg}[C_r]/S_{incr}[C_r] \rfloor$, and the resulting total frame space requested: $B_d := B_r + S_r \times B_s$.

12. If $B_d > B_a$ (there is not enough bandwidth available), proceed to 13. Otherwise (the request can be granted), update $B_{asg}[C_r] := B_{asg}[C_r] + B_r$ (the amount of bandwidth allocated to the traffic class), $B_a := B_a - B_d$ (remaining frame space), $S := S + S_r$ (the number of minislots in the frame). Transmit the scheduling message $< M_r, B_r >$. Then continue at 9 for the next request.

13. There is no bandwidth available to grant the current request. If this request arrived in the last frame (i.e., it is a new request), then transmit the scheduling message $< M_r, 0 >$. Continue at 9 for the next request.

14. Exit from the allocation loop. Calculate the final number of minislots in the frame: $S := min\{S + (B_a/B_s), S_{max}\}$. Transmit $< S, CPF >$ and terminate.

The scheduling algorithm accounts for the CBR sessions in progress, which may temporarily become silent and release some bandwidth. This extra bandwidth can be recycled by non-CBR traffic, but it cannot be allocated to new CBR sessions. The scheduler makes sure that all admitted CBR sessions can always be accommodated into the frame, because, regardless of their intermittent status, they may all become active simultaneously. Therefore, when a new CBR request is being scheduled, the available bandwidth B_a is decremented by B_q, i.e., the amount of bandwidth temporarily released by the silent CBR sessions. This extra bandwidth is included in B_a, and it can be reused by other (non-CBR) sources without problems, as they never make reservations for more than one frame.

Note that although CBR traffic preempts any other traffic class at the bandwidth reservation stage, other traffic classes may take advantage of the silent periods within CBR sessions, which in turn are unavailable to new CBR sessions. Therefore, assuming that the silent periods within CBR sessions are bound to occur, there will always be some spare bandwidth available to lower priority traffic.

After all CBR requests have been processed (step 7), the number of minislots in the new frame is incremented by the number of additional CBR requests that could be accommodated. If no more CBR sessions could be admitted, $CPF[0]$ is cleared to inhibit new CBR requests. Owing to the short due dates of CBR sessions and their relatively long duration, it makes little sense to admit and store new CBR requests if the sessions in progress have used up the entire bandwidth.

2.4.2.2 Determining the number of minislots

When allocating bandwidth to a non-CBR traffic class, the scheduler increases the number of minislots in the frame proportionally to the amount of bandwidth allocated to the class (step 11), to account for the fact that there exist higher priority classes that should be allowed to compete for that bandwidth. This does not happen when bandwidth is assigned to class number 1 (RT-VBR), because the extra minislots for the single class preceding it (CBR) are handled separately (step 7).

Except for that special case, any bandwidth assigned to a lower priority traffic class i is considered available by all classes $< i$. Therefore, such an allocation should result in additional contention opportunities (extra minislots) available to the higher priority classes (appropriately restricted by *CPF*). Note that the number of minislots must be calculated on-the-fly, since the amount of bandwidth available at any moment (B_a) depends on that number. $S_{incr}[i]$, for $2 \leq i \leq C_r$, specifies the amount of bandwidth assigned to class i that will enlarge the contention section by one additional minislot.

The optimum size of the contention section depends on the number of the contending stations and their specific bandwidth requirements. Therefore, although the issue could be studied in more depth, it is impossible to prescribe a single scaling formula that would be optimal in all circumstances. Besides, the absolute accuracy of this formula seems to be of secondary importance from the viewpoint of the overall quality of the discussed access scheme.

With the simple heuristics used in our virtual implementation of the protocol (Section 2.5.2), we assume that we know the average amount of bandwidth $B_v[i]$ allocated to a session in class i. Consider a situation in which some bandwidth B is allocated to an NRT-VBR session (class number 2, according to Table 2.1). From the viewpoint of the RT-VBR class, this bandwidth is available; therefore, it makes sense to increase the size of the contention section by $B/B_v[1]$ minislots—to accommodate that many new RT-VBR contenders. If the same amount of bandwidth B is allocated to a session in class $i > 2$, we should account for the fact that there are several classes considered more important than i, and each of them has its specific average bandwidth requirements. Thus, we use the following harmonic formula:

$$S_{incr}[i] = \frac{1}{\sum_{j=1}^{i-1} \frac{1}{B_v[j]}}, \quad 2 \leq i \leq C_{max}$$

which has the intuitively desirable property that $S_{incr}[2] = B_v[1]$ and $S_{incr}[i] < S_{incr}[i-1]$ for $2 < i \leq C_{max}$.

Note that, instead of being constants, the values $B_v[i]$, $1 \leq i \leq C_{max}$, can be updated on the fly, e.g., using an exponential averaging formula, depending on the measured actual amount of bandwidth allocated to sessions in a given class. In this way, the scaling factors S_{incr} can dynamically adapt to the changing traffic patterns in the network.

2.4.3 Processing at the mobile station

The information broadcast by the base station in the frame announcement packet must be consistently preprocessed by the mobile stations. Fortunately, this processing is straightforward and very inexpensive.

Consider a single selected mobile station receiving an announcement packet. The sequence of scheduling messages $< M_r, B_r >$ arrives before the number of minislots, so the station must wait until the very end of the packet before it can know the exact positions of slots. As a matter of fact, the station does not care about the positions of slots other than its own, which considerably simplifies the processing. The following algorithm lets the station acquire this knowledge.

Scheduling algorithm at the mobile station

Variables:

P	The current position within the frame, that is the next time slot starting position.
P_s	The position of your slot.
P_{max}	The position of the time slot with the maximum length.
B_{max}	The maximum length of a time slot seen so far.
M_r	Indicating the the mobile soure (moile ID).
B_r	Indicating the scheduled bandwidth (slot length).
S	The number of minislots in the frame.
B_s	The amount of frame space needed for one minislot (see section 2.4.2).
P_m	The length of the contention section.

The algorithm

1. Set $P := 0$, $P_s := -1$, $B_{max} := 0$.
2. For all subsequent scheduling messages $< M_r, B_r >$, perform steps 3 through 5, and proceed to 6 when done.

3. If M_r indicates this station, set $P_s := P$, $B_s := B_r$.

4. If $B_r > B_{max}$, set $B_{max} := B_r$, $P_{max} := P$.

5. Set $P := P + B_r$ and continue for the next scheduling message.

6. Read $< S, CPF >$, i.e., the number of minislots and the contention permission flags, and calculate $P_m = S \times B_s$.

7. If $P_s = -1$, terminate. This means that no bandwidth has been allocated to this station. If the request was previously accepted (see step 8), the station must continue waiting until the due date and reissue the request, if it isn't accepted by then. Otherwise, if the request was issued in the previous frame, it didn't make it to the base and must be reissued, according to the contention algorithm (Section 2.4.1).

8. If $B_r = 0$, terminate. This means that the request has made it to the base, but it hasn't been granted in this frame (see step 7).

9. The longest slot is moved to the end. If $P_s > P_{max}$ (our slot is located above the maximum length slot), set $P_s := P_s - B_{max}$; else if $P_s = P_{max}$ (our slot is the maximum length slot), set $P_s := P - B_{max}$.

10. Offset the slot pointer by the amount of space used by the minislots: $P_s := P_s + P_m$.

If all mobile stations execute the same algorithm, they will all come up with the same layout of the new frame, including the location of the longest slot, which is implicitly moved to the very end of the frame. If the frame contains more than one slot with the same maximum length, the first of them (according to the order of scheduling messages in the announcement packet) is moved to the end.

2.4.4 Scheduling UBR traffic

For a traffic class other than UBR, the *requested bandwidth* specified by the source in its request packet is treated by the scheduler as the exact amount to be allocated. Thus, it directly corresponds to B_r, as used in the scheduling algorithm discussed in Section 2.4.2. The interpretation of the requested bandwidth for UBR traffic is slightly different. The scheduler assumes that a UBR source can sensibly use any bandwidth whatsoever, and the specification arriving in the request packet is viewed as the maximum.

Several fair scheduling strategies for UBR traffic are possible. With the simplest strategy (requiring the least amount of work from the scheduler), the outstanding UBR requests

are processed in a round-robin fashion, and each of them receives as much bandwidth as available within the currently scheduled frame, up to the requested maximum. This approach, dubbed the *least effort policy* (LEP), minimizes the fragmentation of UBR packets into slots (individual UBR slots are as large as possible), and thus maximizes the throughput (frame utilization), but incurs the longest delays. Depending on the true nature of the UBR applications, and the number of active UBR mobiles, this delay may be acceptable or not. Generally, due dates for UBR requests can be set very late, because there are no inherent timing constraints for this type of service.

On the other end of the spectrum is the *maximum population policy* (MPP), which maximizes the number of mobiles that can be serviced within one frame. With this approach, UBR slots will tend to be shorter (incurring more bandwidth overhead), but the delays perceived by individual users will be shorter as well.

The entire spectrum can be described by a single policy parameterized by the *preferred granularity* of bandwidth allocation, which we shall denote G_{ubr}. The idea is to allocate the UBR bandwidth in chunks of G_{ubr} cells, up to the amount requested by the source. Any leftovers are first spread among the sources that have already received bandwidth (but still need more), and then assigned to the remaining stations with the granularity as close to G_{ubr} as possible. If $G_{ubr} = \infty$, the resulting policy is LEP, if $G_{ubr} = B_{min}$, where B_{min} corresponds to the amount of bandwidth needed to accommodate a single ATM cell, we get MPP.

2.5 Virtual implementation

In this section, we describe the simulation model of DS-TDMA/CP that we used to study the performance of our proposed scheme. The numerical parameters assumed in this model, especially the network parameters, can be viewed as suggested values for a possible real-life implementation.

2.5.1 Traffic models

To study the performance of DS-TDMA/CP, we consider three traffic classes: CBR, which receives a special treatment and therefore must be included in any study, VBR, as the highest-priority non-CBR traffic pattern, and UBR exemplifying data traffic with the lowest priority. The CBR traffic is assumed to be voice, the VBR traffic represents video, and the UBR traffic corresponds to file transfers, i.e., data, without stringent delay requirements.

2.5.1.1 CBR traffic

Voice traffic is commonly represented with an "On-Off" model, as described in section A.3.1. DS-TDMA/CP views a CBR source as being in one of four states (see Figure 2.3): *Idle* (I), *Request* (R), *Talking* (T), and *Silent* (S). When the user initiates a call, the mobile station transits from *Idle* to *Request* and issues a bandwidth reservation request to the base station. The mobile remains in the *Request* state until it is assigned a channel (i.e., a time slot) by the base station, or until its request reaches the due date and expires. In the latter case, the mobile will become *Idle* again and the user will receive a busy signal.

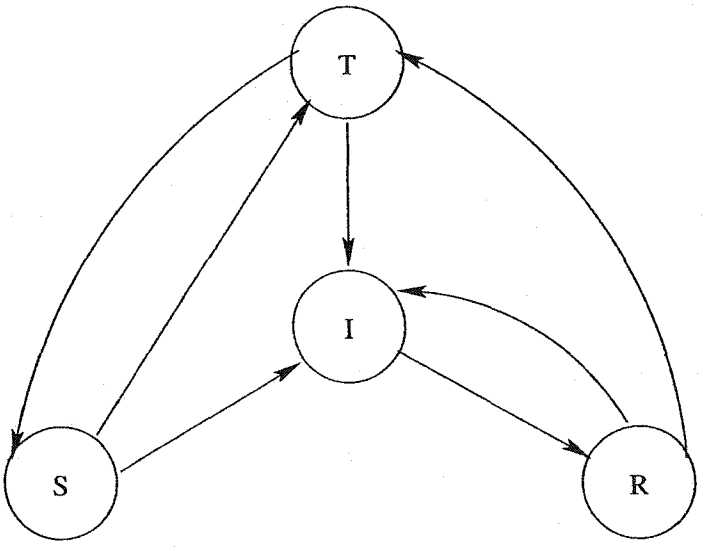


Figure 2.3: The CBR model

When the request of the mobile station is granted, a CBR session is set up between the mobile and the base. The mobile will be generating talkspurts interleaved with silence periods for as long as the session is not explicitly torn down.

2.5.1.2 VBR traffic

For the VBR video traffic, we adopt the DAR(1) (Discrete Autoregressive) model as described in Section A.3.2. Similar to a CBR source, a VBR source can be in one of four states: *Idle* (I), *Request* (R), *Scheduled* (S), and *Waiting* (W). For as long as the VBR mobile has nothing to transmit, it is in the *Idle* state. When a new video frame is generated and the buffer is empty, the mobile issues a bandwidth request to the base station and enters the *Request* state. The station remains in the *Request* state until its request is fulfilled, and then it enters the *Scheduled* state to transmit packets in the allocated time slot.

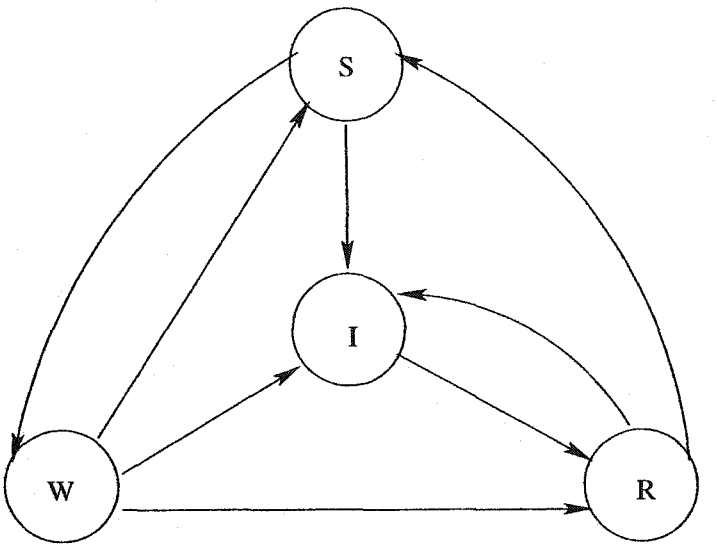


Figure 2.4: The VBR model

If the mobile times out while in the *Request* state, it drops the current frame and transits to *Idle*, but only if the buffer is empty. Otherwise, it remains in the *Request* state to issue requests for subsequent video frames.

If the mobile finishes its transmission, there are two possibilities. If the buffer was nonempty when the transmission was started (and the station piggybacked its new request in the transmitted slot), it transits to *Waiting* to wait for a new slot to be scheduled by the base. If the buffer was empty, the mobile transits to *Idle* where it will issue a new request as soon as a new frame appears in the buffer.

While in the *Waiting* state, the mobile may transit to *Scheduled*, if it receives a slot on time, or to *Idle*, if the request times out and there are no more frames in the buffer. Figure 2.4 illustrates the transitions among those four states.

2.5.1.3 UBR traffic

The UBR traffic is assumed to represent non-critical applications, which pose no stringent delay requirements and expect no special quality of service other than best-effort delivery. Consequently, in our model, although an UBR session may expire while waiting for access to the base station (the source may lose its patience), once the request has made it to the base, it will never be dropped, i.e., its deadline is assumed to be infinite. UBR sessions can be viewed as file transfers.

Like a VBR source, a UBR source can be in one of four states: *Idle* (I), *Request* (R),

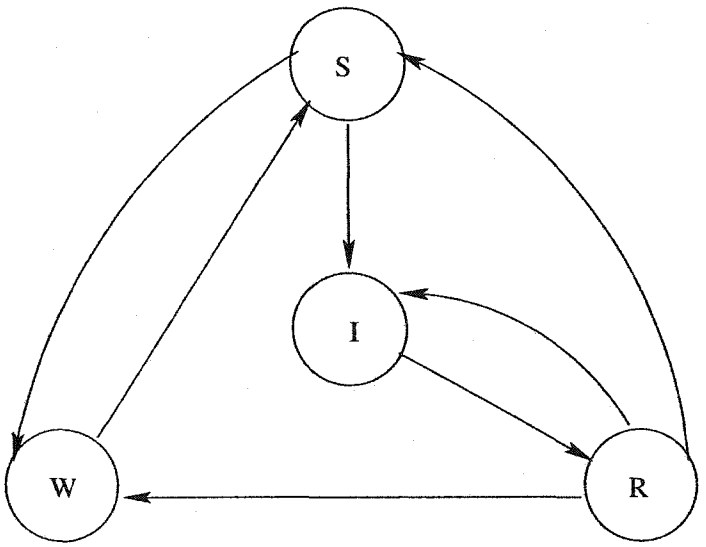


Figure 2.5: The UBR model

Scheduled (S), and *Waiting* (W)—see Figure 2.5. When there is a file to be transmitted, the mobile transits from *Idle* to *Request* and issues an access request specifying the maximum amount of bandwidth that the station could possibly use within one frame. As explained in Section 2.4.4, this specification is viewed by the bandwidth scheduler as a maximum, and the base station is free to grant the mobile whatever bandwidth it considers appropriate. The mobile remains in the *Request* state until its request is received by the base station. If the base responds with a nonzero assigned bandwidth, the mobile transits from *Request* to *Scheduled* to transmit a portion of its outstanding data.

If the base has received the request but the allocation is delayed (i.e., the base responds with zero slot length), the mobile transits to state *Wait*.

For as long as there remain data to be sent, the source will piggyback new bandwidth requests onto the transmitted packets and transit form *Scheduled* to *Wait* at the end of every transmission. Having completed the transmission of the last packet of its current file, the mobile transits to *Idle* to await the arrival of another file to transmit.

2.5.2 Numerical parameters of the model

We consider a network consisting of a single base station and a variable number of mobiles (up to 100 in each traffic class) possibly trying to talk to the base at the same time. The transmission rate of the channel is 1Mb/s. The propagation delay of the channel is ignored; however, its impact is captured by the guard time included in the length of every slot and minislot. For uniformity, we express all parameters in bits, assuming that 1 bit = 1μs.

With reference to Figure 2.1 and Section 2.4.2, the exact numerical parameters assumed in our experimental setup are listed in Table 2.2. The 60 bit request packet length results from putting together the packet preamble (32 bits), service type (3 bits), user identification (12 bits), requested slot length (8 bits), and the due date (5 bits). Note that the slot length is quantized into ATM frames. Similarly, the due date is represented as 32 discrete levels that, in real life, can be transformed through a lookup table (or some possibly nonlinear formula) into whatever deadlines are considered sensible and natural for the collection of traffic classes handled by the network.

frame length (B)	60,000
slot guard time (GT)	8
request packet length (B_s)	60
maximum number of minislots (S_{max})	856
preamble length (PA)	32
slot header length (SH)	2 – 18
packet serial number (PSN)	8
ATM VPI *and* VCI	16+16
CRC + FEC	8+8
ATM *payload*	384

Table 2.2: Network parameters

The length of the slot header (SH) field, which is used for piggybacking bandwidth requests, depends on the traffic class to which the slot has been allocated. For a CBR session, SH merely indicates the session state (active, silent, hung up—see Section 2.3), which information can be accommodated into 2 bits. Note that even if we admit CBR sessions with different bandwidth requirements, the bandwidth (originally specified in the contention minislot) remains the same for the entire session. By definition of a CBR session, there is no need to piggyback detailed bandwidth specification in CBR packets. For VBR and UBR sessions, the piggybacked bandwidth request must include a slot size and due date, i.e., the SL and DD fields from the request packet. Thus, for these traffic classes, the length of the SH field is 12 bits.

The slot length for an active CBR session (a talkspurt packet) is set to 2154 μs. The bit rate of a CBR session is assumed to 32 kb/s, with the mean durations of the talkspurt and silence periods settable as simulation parameters. Ignoring the overhead, a single CBR session requires 32/1000 of the channel time during its talkspurt phase, which translates into the same fraction of every frame. The CBR slot length includes that fraction as well as various overheads needed to turn the talkspurt frame into a transmittable unit.

The variance and auto-correlation parameters for the VBR traffic model (Section 2.5.1.2)

have been set to 5536 and 0.98, respectively [58]. The due date for all VBR packets is 150 ms. The mean source rate of a VBR session is assumed to be 128 kbps/s. The mean duration of a single VBR call is three minutes.

A UBR session represents a file transfer and, once the request makes it to the base station, its due date is infinite. However, a UBR request may expire while the source is waiting for access to the base station (i.e., permission to contend and a scheduling message). The expiration time, after which a UBR request is dropped at the source, has been set at 30 seconds. The mean length of a file to be transferred is 850 kb. The transmission rate is flexible: the source can use whatever bandwidth is left over within the frame after the two higher priority traffic classes have been serviced. Note that because CBR sessions occasionally become silent, and VBR (as well as CBR) sessions have more rigid requirements regarding the slot size than UBR sessions, there usually is some bandwidth left for UBR traffic, even if the network is heavily loaded by CBR/VBR traffic. The G_{ubr} (preferred granularity) parameter for scheduling UBR traffic (Section 2.4.4) has been set at three ATM cells.

2.6 Performance

The performance of DS-TDMA/CP has been investigated by simulation and compared to the performance of D-TDMA and DQRUMA—two representatives of the TDMA family offering most promising means of accommodating multiple traffic classes with different patterns and QoS expectations. We have analysed the proposed solution with respect to the following performance aspects:

QoS trade-offs. DS-TDMA/CP purports to cater to traffic classes with different priorities and QoS requirements. We would like to see how those multiple traffic classes share the network bandwidth and whether their actual handling by the protocol in terms of relative performance matches our intentions.

Responsiveness and flexibility. The primary idea behind delaying the scheduling decisions until the last possible moment is to accommodate as many requests as can be sensibly accommodated within the maximum delay of one frame. We would expect our protocol to be highly responsive to intermittent changes in load patterns and flexible with bandwidth allocation.

Bandwidth utilization. Owing to the variable slot size, DS-TDMA/CP should offer better effective throughput (i.e., smaller overhead) than solutions based on fixed size slots.

2.6.1 QoS trade-offs

2.6.1.1 Balanced load

The single most illustrative performance graph visualizing the properties of DS-TDMA/CP at a glance is Figure 2.6. The figure shows the fraction of the network's effective bandwidth used by each of the three traffic classes under increasing load conditions. The *load factor* of 0.8 indicates the average fraction of all mobiles (their total population is marked on the x-axis) involved in traffic sessions of the indicated type. The number of mobile stations from each traffic class is set to be the same.

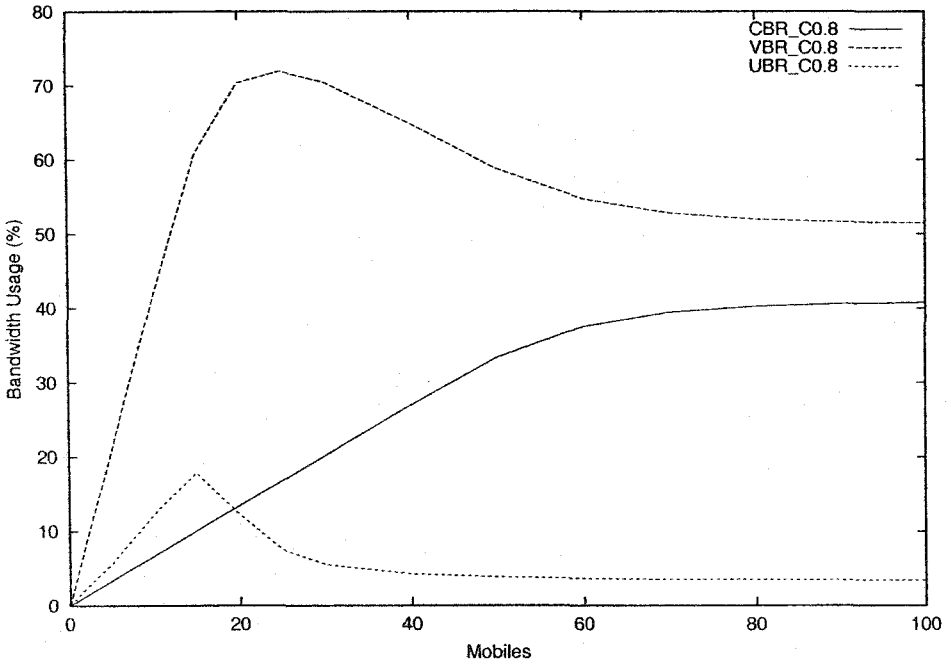


Figure 2.6: Bandwidth utilization in DS-TDMA/CP

At the beginning, when the number of mobiles is small, there is enough bandwidth to accommodate all sessions without preemption. Then, the relative positions of the curves indicate the individual contributions of the three traffic types to the total offered load.

When the system load reaches a certain threshold (15 mobiles, about 88% capacity), the service received by UBR sessions begins to decline—to make room for CBR and VBR traffic. This is because UBR traffic has the lowest priority and always yields to CBR and/or VBR. Note, however, that the system does not provide its maximum capacity at this point. Because all VBR and CBR sessions are satisfied, some bandwidth is set aside for additional contention minislots, as described in Section 2.4.2. This mode of operation is sustained for as long as all VBR requests are still satisfied. Having a higher priority than UBR requests,

they are first to reuse the silent slots left over by the CBR sessions in progress. The system reaches its maximum capacity around 25 mobiles, when some VBR sessions begin to be rejected in order to satisfy the increasing number of CBR requests.

With DS-TDMA/CP, the maximum number of CBR mobiles C_{CBR} that the system can support is limited by the frame size (B), the amount of bandwidth reserved for minislots ($S \times B_s$), the slot size assigned to an active CBR session ($B_v[0]$), and the *load factor* of a CBR mobile (C). Formally, this limitation can be expressed as

$$C_{CBR} = \frac{B - S\,B_s}{B_v[0]\,C} \quad . \tag{2.1}$$

When the number of CBR mobiles in the system is over C_{CBR}, some CBR access requests will be blocked, as shown in Figure 2.7 (in this case $C = 0.8$ and $C_{CBR} = 40$).

Comparing Figure 2.7 to Figure 2.6, we can see that the CBR bandwidth usage is still going up even after the population of CBR mobile has exceeded C_{CBR}. This results from the statistical impact of the *load factor* C. When $C = 0.8$, not all CBR mobiles are active simultaneously all the time. Sometimes the number of active CBR sessions is over C_{CBR} and some CBR access requests are blocked; sometimes, the number of active CBR sessions is below C_{CBR}. On the average, the active CBR sessions don't reserve all the bandwidth all the time. These fluctuations also explain why the VBR traffic does not reach its minimum throughput even when some CBR sessions begin to be blocked.

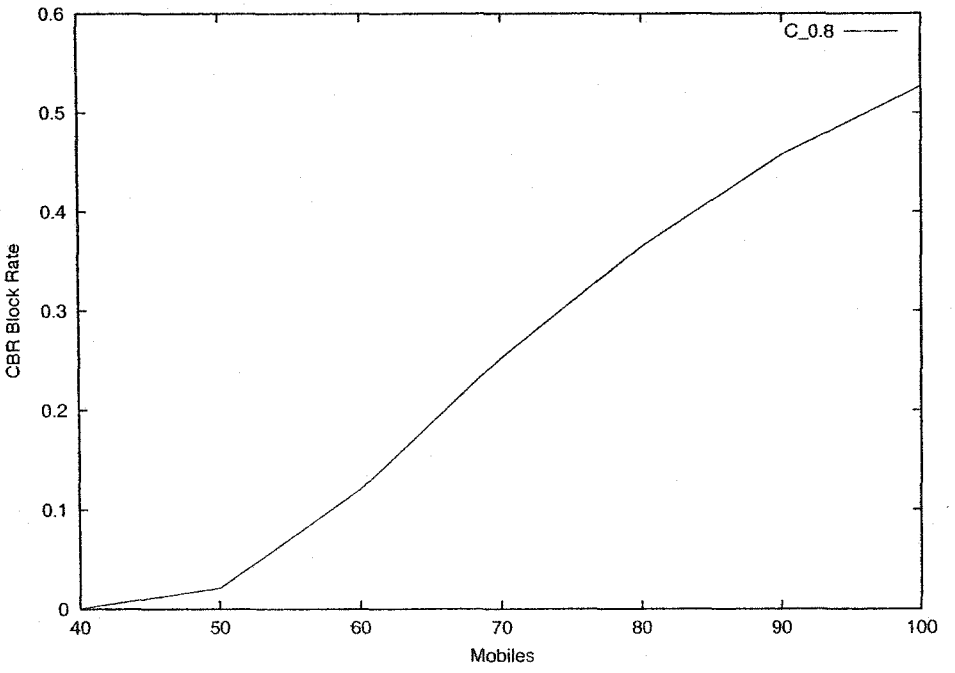


Figure 2.7: CBR blocking rate in DS-TDMA/CP

The amount of CBR traffic accommodated by the system increases steadily with the increasing load from CBR sessions until its specific saturation threshold. The other two traffic types, i.e., VBR and UBR, yield to CBR, with UBR additionally yielding to VBR, although neither of them dies out. In fact, VBR stabilizes at a level that is considerably higher than the saturation threshold of CBR. This is because, as we explained in Section 2.4.2, CBR traffic is incapable of using more than a certain fraction of the entire useful bandwidth, roughly equal to $l_t/(l_t + l_s)$, where l_t and l_s are the average lengths of the talkspurt and silence periods. Similarly, UBR traffic is not completely eliminated by VBR. This is because VBR packets have much stricter bandwidth requirements than UBR packets, which can use practically any leftovers. In consequence, there is always some bandwidth unusable by VBR sources but still allocatable to the less picky UBR sources.

The distribution of bandwidth in DS-TDMA/CP has also been studied under different load factors and different distributions of the silent/active periods in CBR sessions. Figure 2.8 shows the impact of the load factor on the amount of bandwidth used by each traffic class. The bandwidth utilization curves in Figure 2.8 have similar shape to those in

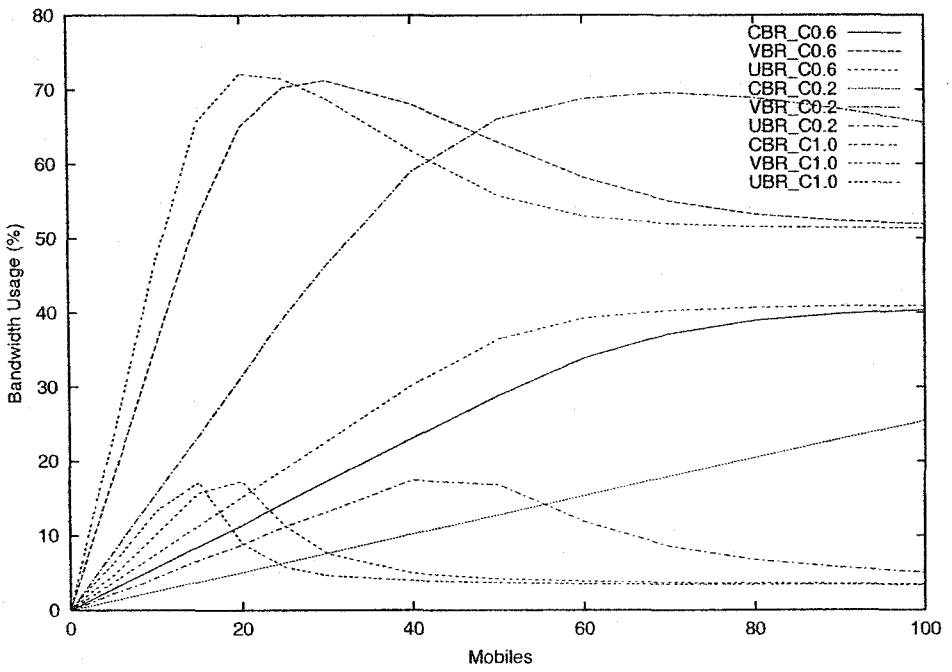


Figure 2.8: Bandwidth utilization in DS-TDMA/CP, $C = 0.2, 0.6, 1$

Figure 2.6. The difference is that VBR and UBR traffic classes can get more bandwidth when the load factor is small because of the reduced activity of CBR mobiles.

Figure 2.9 shows the impact of varying the silent/active time ratio in CBR sessions on the

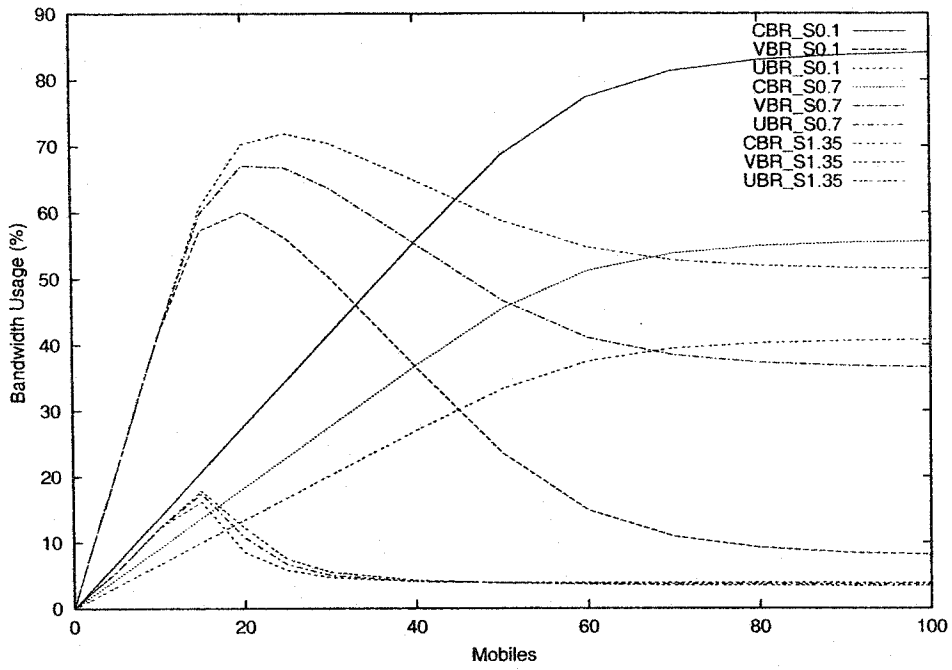


Figure 2.9: Bandwidth utilization in DS-TDMA/CP under different silent/active ratios

bandwidth utilization among the three traffic classes. S0.1, S0.7 and S1.35 correspond to the silent/active ratios of 0.1:1, 0.7:1 and 1.35:1, respectively. The bandwidth utilization follows the same pattern regardless of the silent/active ratio, but more bandwidth is available for VBR and UBR sessions when silent/active ratio is high. This is simply because more bandwidth can be reused by those sessions from the inactive periods in CBR sessions.

Although D-TDMA exhibits a bandwidth usage pattern similar to DS-TDMA/CP (see Figure 2.10), the CBR throughput curve has a steeper slope than in Figure 2.6. This is because the frame structure in D-TDMA is rigid and constrained by the channel rate and the source rate. Moreover, the CBR sessions in progress never make their silent periods available to other traffic classes.

Notably, the maximum used CBR bandwidth reaches about the same level as in DS-TDMA/CP. This is not surprising because the frame structure in D-TDMA was tailored specifically for CBR traffic. However, the effectively used portion of the CBR bandwidth is about 18%, as shown by the dotted CBR curve that excludes the non-reusable silent periods. For this reason, the bandwidth assigned to VBR traffic (and the total useful bandwidth of the network) ends up at a considerably lower level than in DS-TDMA/CP.

In DQRUMA (Figure 2.11), the rigid bandwidth allocation scheme limits the system performance in a completely different way. According to the protocol, access requests from

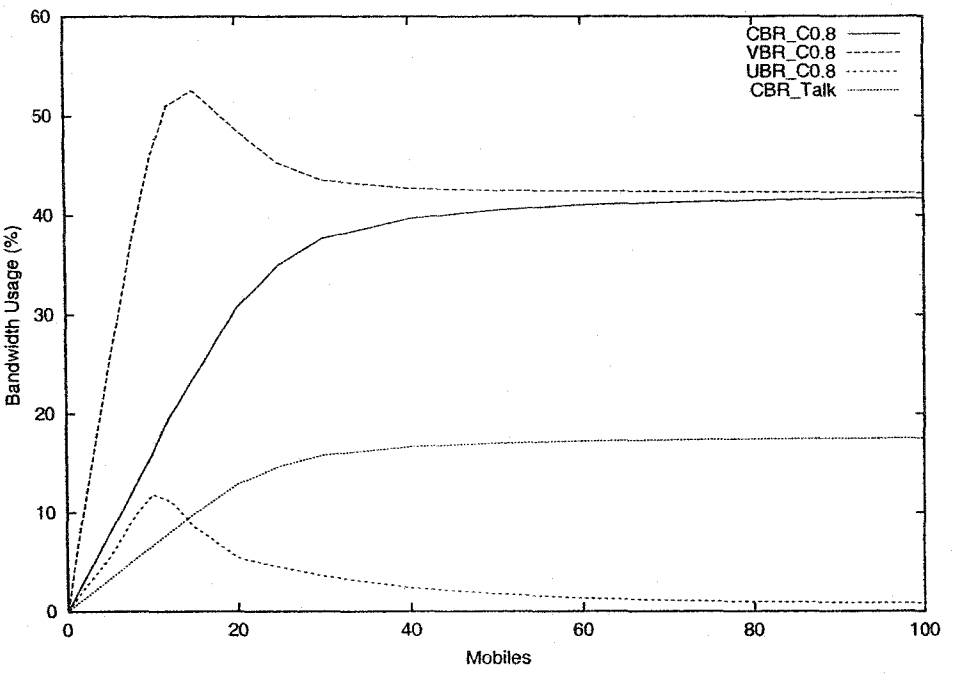


Figure 2.10: Bandwidth utilization in D-TDMA

all kinds of traffic are scheduled in a round-robin fashion. In our experiment, VBR is the traffic class with the highest number of individual requests. Consequently, within the light range of traffic conditions, when the system has enough bandwidth to accommodate all requests, VBR is getting more bandwidth than the other two classes. With the increasing load (from all classes), the round-robin policy tends to incur longer and longer scheduling delays. Some CBR and VBR requests (for which deadlines matter) time out and become dropped. The increased number of mobiles contending for their first transmission (and unable to piggyback their requests), push up the collision rate, which further reduces the amount of usable bandwidth. Since the VBR class has the highest number of requests, it is most adversely affected by this behavior, and the VBR bandwidth drops first. CBR sessions, needing fewer access requests to receive their share of bandwidth, are affected to a lesser extent. With the increasing number of CBR requests, the bandwidth allocated to voice traffic continues to grow for a while after VBR has reached its maximum. Although the UBR traffic is also affected by the poor accessibility of bandwidth, those requests never time out. As long as an UBR request makes it to the base, it will be serviced at some point, and once that happens, the session will be able to sustain itself via the piggyback mechanism. This is why the bandwidth used by the UBR class keeps increasing to the very end of the investigated range of traffic conditions.

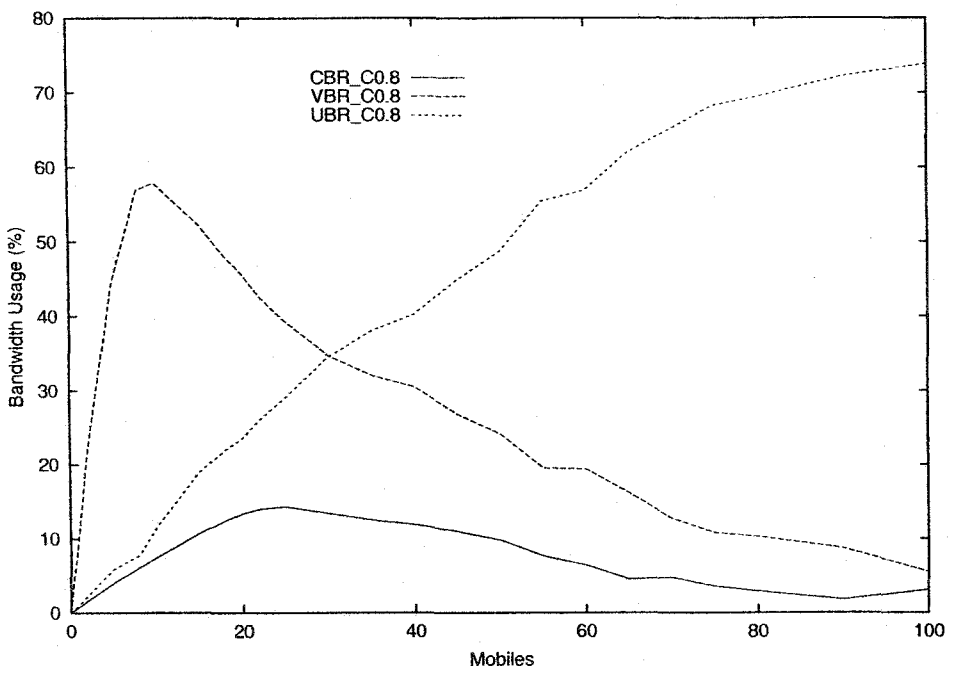


Figure 2.11: Bandwidth utilization in DQRUMA

2.6.1.2 Biased load

To demonstrate that the QoS received by CBR sessions in DS-TDMA/CP is not affected by the presence or absence of other traffic types, we carried out a series of experiments in which the offered CBR load remained steady, while the contribution of other traffic classes varied. The CBR load was set at a high level—to make the drop rate (and any deviations of it) clearly visible, while the VBR and UBR load increased in proportion to the number of mobiles in the network. Figure 2.12 illustrates the stability of the CBR service in such circumstances for three different (two of them high) fixed levels of CBR load. The blocking rate is always a straight horizontal line,[5] which demonstrates that the CBR traffic has absolute priority in acquiring bandwidth.

Another illustration of the quality of service received by different traffic classes in DS-TDMA/CP is given in Figure 2.13. It shows the bandwidth utilization in a network under traffic conditions similar to those shown in Figure 2.6, except that the UBR load is kept constant at the highest level. The curves for VBR and CBR are completely indistinguishable from those in Figure 2.6.

The influence of heavy CBR traffic on the throughput of VBR traffic is shown in Figure 2.14, which was obtained under conditions similar to those depicted in Figure 2.6, except

[5]The blocking rate for the load level of 0.2 is consistently zero.

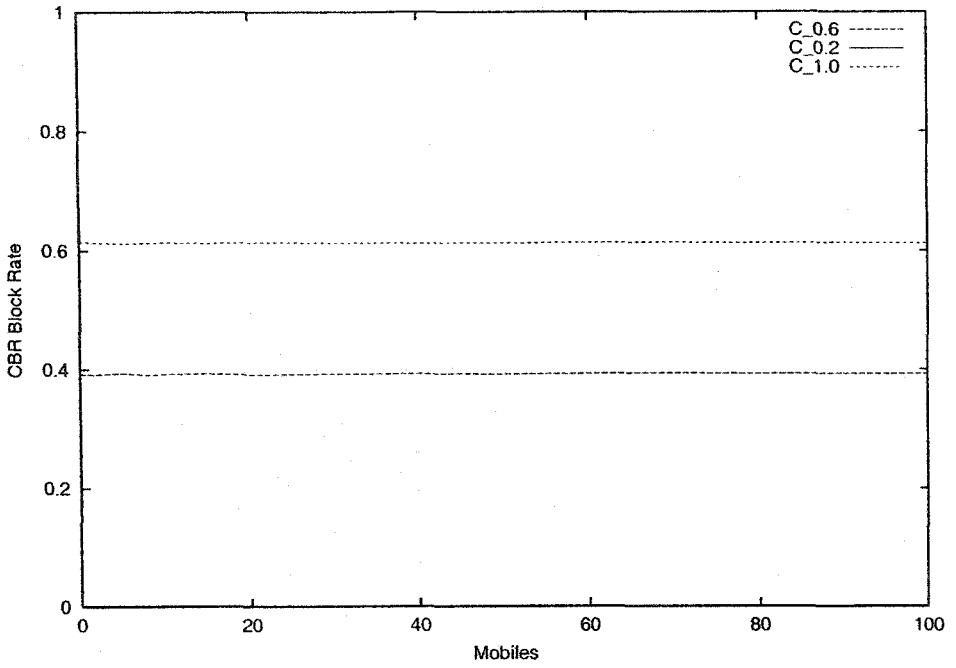


Figure 2.12: Blocking rate in DS-TDMA/CP for CBR traffic under constant CBR load

that the CBR load was set to a very high constant level. In this case, VBR can only receive bandwidth from the silent periods in CBR sessions, and the hump in the VBR curve disappears.

When the VBR traffic is dominant, UBR sessions receive little bandwidth, especially when the CBR and UBR load is light—as shown in Figure 2.15. This is because, the heavy VBR traffic consumes most of the bandwidth, and there is little bandwidth from the silent periods in CBR session that could be reused by VBR/UBR,

Ignoring the numerical differences in achievable throughput, the behavior exhibited by D-TDMA under such conditions is very similar to DS-TDMA/CP, which is more than can be said about DQRUMA. Figure 2.16 was obtained under fixed heavy CBR load, with the remaining traffic classes behaving as in the experiment illustrated in Figure 2.11. Owing to the round-robin bandwidth allocation policy, the performance of DQRUMA for high-priority traffic is severely impaired by the presence of low-priority load. With the increasing UBR and VBR loads, the CBR traffic receives consistently smaller and smaller share of the network bandwidth. This happens because CBR sessions, having short deadlines, also have hard time getting through the contention stage. Consequently, they tend to time out and be dropped more often than other requests.

In fact, somewhat contradictory to its purpose, DQRUMA seems to favor UBR traffic,

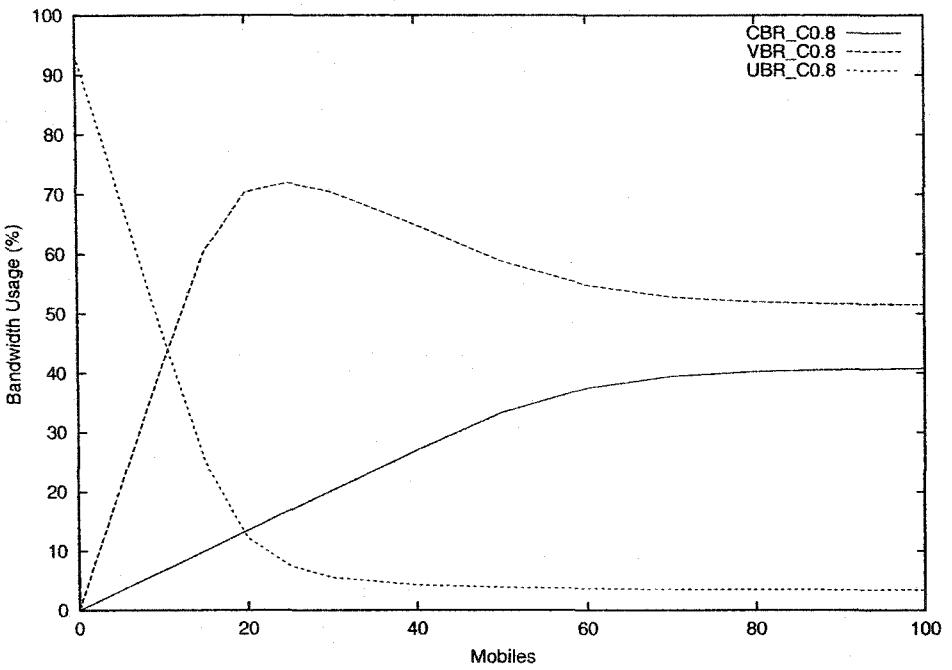


Figure 2.13: Bandwidth utilization in DS-TDMA/CP under constant UBR load

or, for that matter, any traffic class with long or infinite deadlines. Under all conditions, the amount of bandwidth allocated to UBR sessions tends to grow until it takes most of the available bandwidth. Any request whose deadline is sufficiently long is eventually able to make it to the base, and once that happens, the session will sustain itself through the piggyback mechanism of DQRUMA.

2.6.2 Burst responsiveness

Even though in the long term (at equilibrium) D-TDMA treats prioritized traffic classes in a way similar to DS-TDMA/CP, both D-TDMA and DQRUMA show a significantly slower responsiveness to rapid fluctuations in the offered load. This is illustrated in Figures 2.17–2.19. All three figures illustrate a sudden transition of the network from a completely idle state to being heavily loaded with traffic of all three classes (all sessions start up at the same time). The average duration of the talkspurt phase for a CBR session is equal to 1 second.

Owing to the inherently dynamic behavior of DS-TDMA/CP, its high responsiveness to bursts is somewhat disguised in Figure 2.17. Because of the relaxation of bandwidth by the silent CBR sessions, any short time bandwidth utilization curve obtained under heavy CBR load is going to show fluctuations. Another confusing factor is the dynamic layout of the uplink frame, with its variable-size contention section, which also impacts the

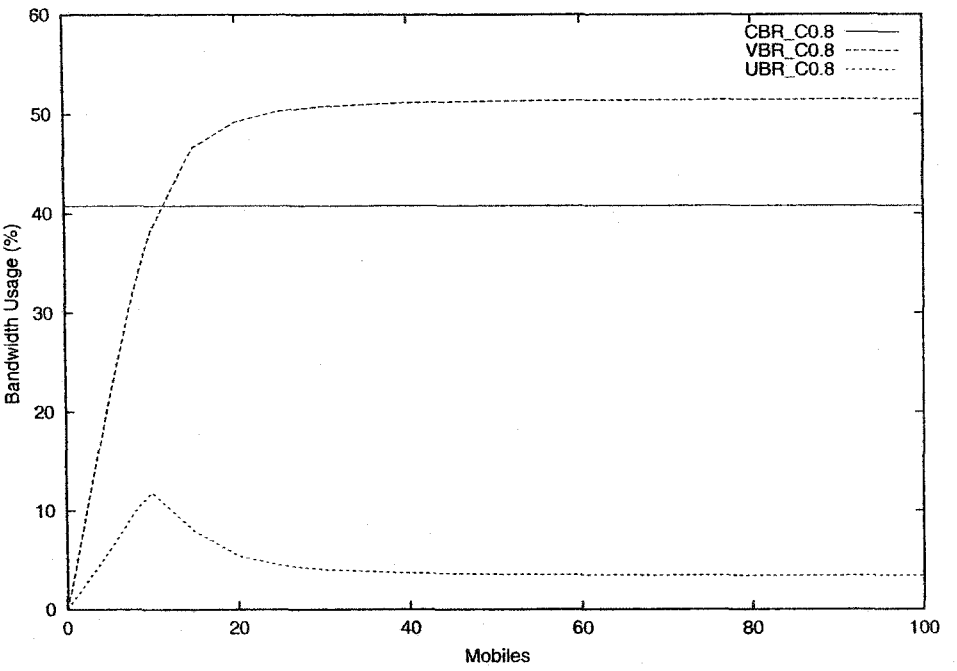


Figure 2.14: Bandwidth utilization in DS-TDMA/CP under constant CBR load

responsiveness of the collision resolution scheme.

Thus, the bandwidth used by the CBR class appears to drop for a while, which reflects the fact that some of the sessions switch to the silent phase (a CBR session always starts in a talkspurt). Note the initial flat portion of the CBR curve in Figure 2.17, which corresponds to the period during which all CBR sessions are still active. Within that period, the network behavior is stable from the very beginning, which is not the case with the remaining two protocols. Both D-TDMA and DQRUMA exhibit a considerably unstable behavior during an initial period of the burst.

The layout of the contention section of a frame in D-TDMA is fixed. Consequently, a considerable amount of bandwidth is wasted at the initial stage of burst resolution on coping with the contention among a huge number of requests arriving all the same time. DQRUMA suffers from a similar problem. Although, like DS-TDMA/CP, it tries to be flexible with the number of contention minislots, owing to the significantly shorter frame in DQRUMA, this number turns out to be inadequate for a quick resolution of the burst. Moreover, according to what we observed in Section 2.6.1.1, once the network recovers from the burst, UBR takes most of the bandwidth, and CBR ends up with the least share. In contrast, when a large number of requests show up in an idle network driven by DS-TDMA/CP, the contention opportunities are much better than in DQRUMA and D-TDMA, and the burst is resolved

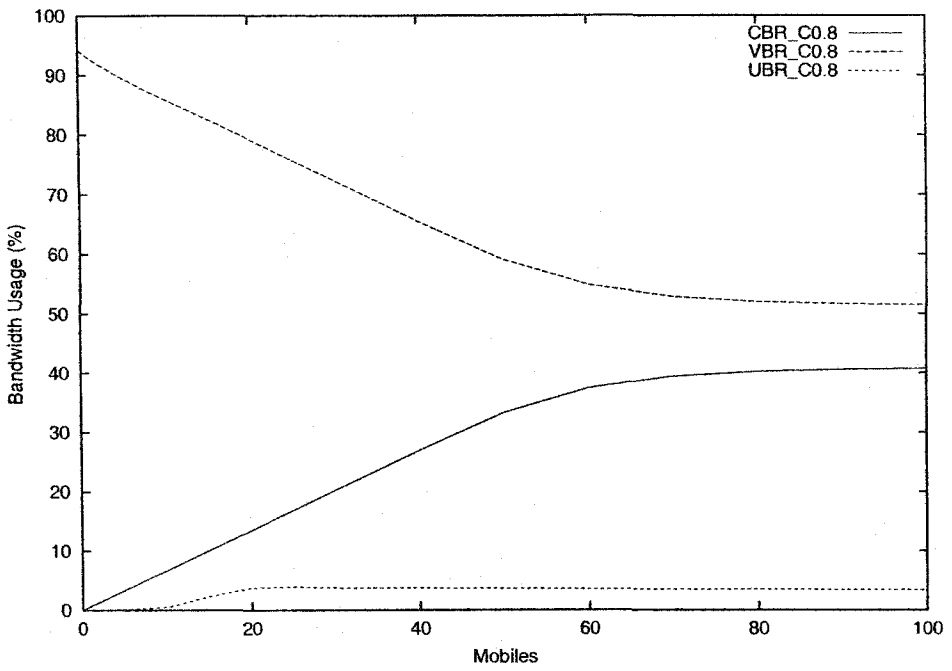


Figure 2.15: Bandwidth utilization in DS-TDMA/CP under constant VBR load

incomparably faster.

We can clearly see the perfectly stable behavior of DS-TDMA/CP from the very beginning of the burst in Figure 2.20. This graph has been obtained under conditions similar to those depicted in Figure 2.17, except that the duration of the talkspurt phase was extended to 100 seconds—to eliminate the impact of CBR sessions becoming silent on bandwidth fluctuations during burst resolution.

2.6.3 Overhead

In comparing the bandwidth overhead incurred by the three protocols, we consider three mixes of traffic and three different bit rates for CBR sessions. Note that parameters like frame and slot size (for the fixed slot size protocols) are constrained by the channel rate, CBR rate and the CBR slot size.

Each of the three traffic mixes strongly favors one traffic type. With the mix denoted M_{CBR}, 90% of all traffic is contributed by CBR sessions, while the remaining traffic types (i.e., VBR and UBR) contribute 5% each. Similarly, mixes M_{VBR} and M_{UBR} favor VBR and UBR traffic types—in the same proportions. Reasonably accurate approximations of the overhead for other mixes can be obtained from the presented numbers by straightforward interpolation.

Table 2.3 lists the bandwidth overhead measured for the three investigated protocols

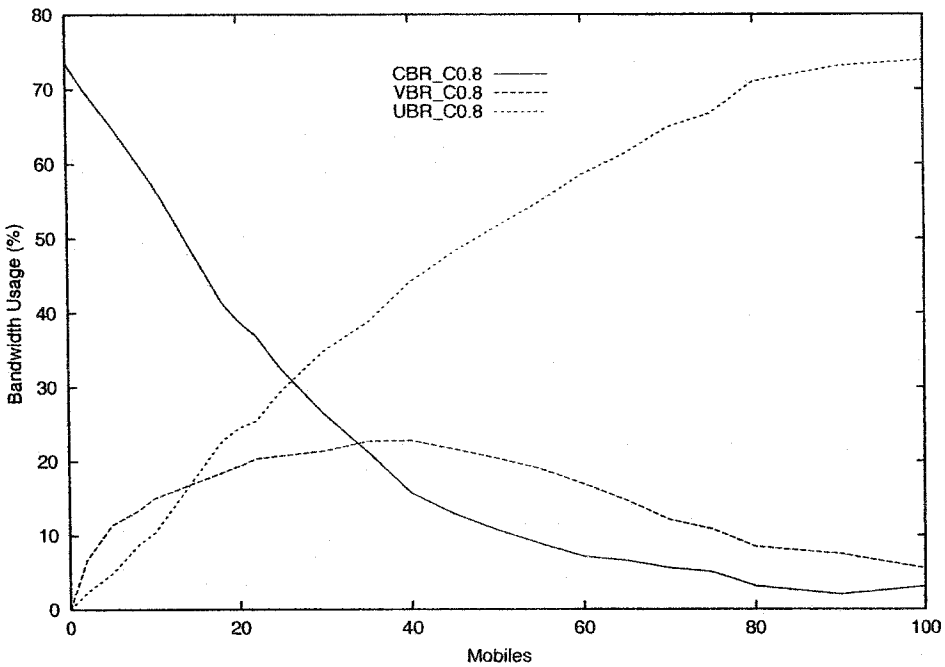


Figure 2.16: Bandwidth utilization in DQRUMA under constant CBR load

Protocol	CBR rate = 16kbps			CBR rate = 32kbps			CBR rate = 64kbps		
	M_{CBR}	M_{VBR}	M_{UBR}	M_{CBR}	M_{VBR}	M_{UBR}	M_{CBR}	M_{VBR}	M_{UBR}
DS-TDMA/CP	14.77	6.51	5.28	9.96	7.11	5.14	10.74	6.88	4.86
D-TDMA	15.18	12.22	12.22	15.21	15.21	15.20	17.68	16.26	15.20
D-TDMA*	50.53	18.36	18.88	39.49	24.58	24.58	41.90	31.90	30.84
DQRUMA	17.36	17.40	17.35	17.38	17.41	17.35	17.35	17.40	17.35

Table 2.3: Bandwidth overhead for 1Mb/s channel

under three traffic mixes and three CBR rates. The row labeled D-TDMA* includes the overhead caused by the silent periods in CBR traffic, which are unusable in D-TDMA. The overhead is calculated as the percentage of bandwidth of the uplink channel not used to transmit any data bits. Thus, it covers all guard intervals, gaps, headers and trailers. Table 2.4 lists the overhead measured for three different channel rates, assuming that the CBR session rate is 32kbps.

Notably, for the traffic mix with the predominant CBR component, the overhead incurred by DS-TDMA/CP is not much lower than for the other protocols, [6] although it tends to decrease with the increasing rate of CBR. The situation changes quite drastically, when the network load becomes dominated by non-CBR sessions. Note that for higher channel rates, the overhead of DS-TDMA/CP additionally tends to decrease, which demonstrates

[6]Which have been designed with the CBR traffic in mind.

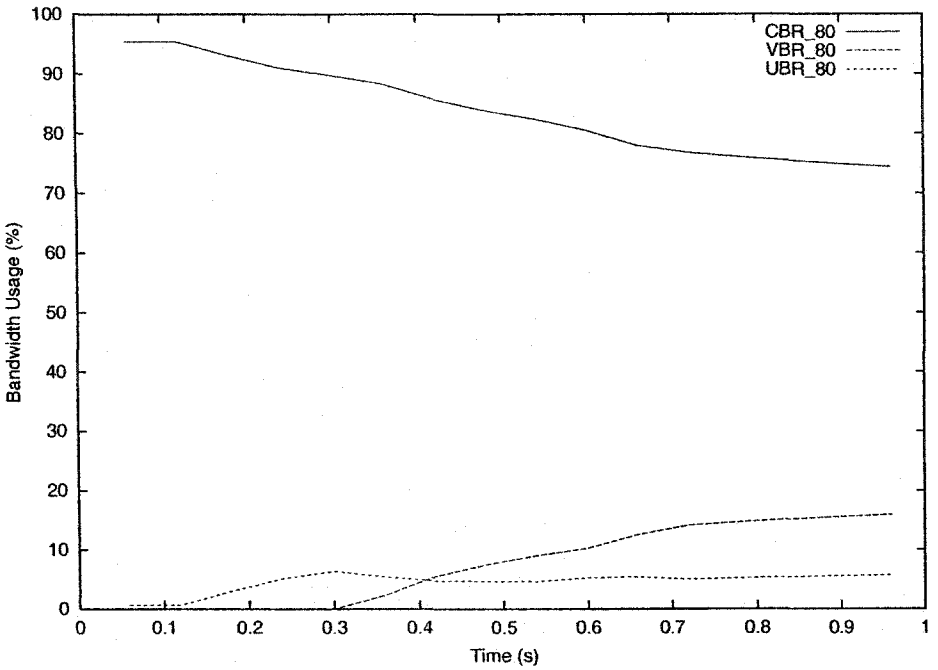


Figure 2.17: Burst response of DS-TDMA/CP, short talkspurt, heavy load

Protocol	channel rate = 0.5Mbps			channel rate = 1Mbps			channel rate = 1.5Mbps		
	M_{CBR}	M_{VBR}	M_{UBR}	M_{CBR}	M_{VBR}	M_{UBR}	M_{CBR}	M_{VBR}	M_{UBR}
DS-TDMA/CP	8.45	12.68	8.62	9.96	7.11	5.14	17.19	4.14	3.90
D-TDMA	15.20	15.33	15.20	15.21	15.21	15.20	15.23	15.20	15.20
D-TDMA*	39.53	30.93	31.03	39.49	24.58	24.58	39.48	21.45	21.45
DQRUMA	17.67	17.59	17.35	17.38	17.41	17.35	17.46	17.37	17.35

Table 2.4: Bandwidth overhead for different channel rates

that the protocol scales well to the increasing transmission rate of the mobile network.

2.7 Conclusions

DS-TDMA/CP offers differentiated quality of service to multiple traffic classes demanded by contemporary mobile applications. Owing to the variable and flexible slot size, this protocol incurs a lower bandwidth overhead than other solutions based on fixed size slots, especially when CBR traffic is not the dominant load in the network. By deferring scheduling decisions until the last possible moment and greatly improving contention opportunities under light load, the proposed access scheme is highly responsive to rapid changes in the traffic pattern. This feature also makes it possible to identify and reuse silent periods in voice sessions to accommodate other traffic, without forcing the voice sessions to contend for bandwidth again at the beginning of a new talkspurt.

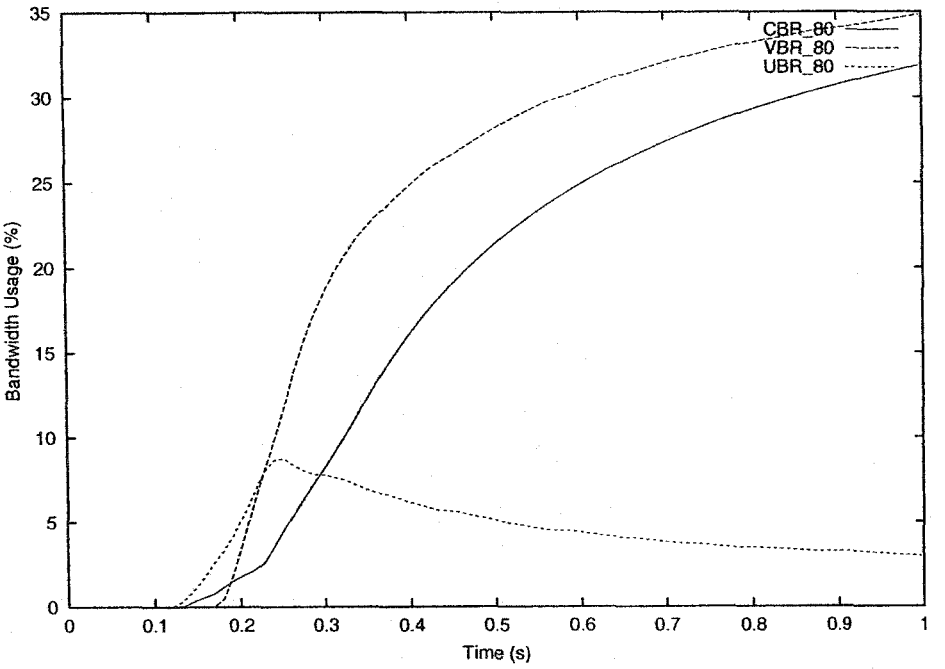


Figure 2.18: Burst response of D-TDMA, heavy load

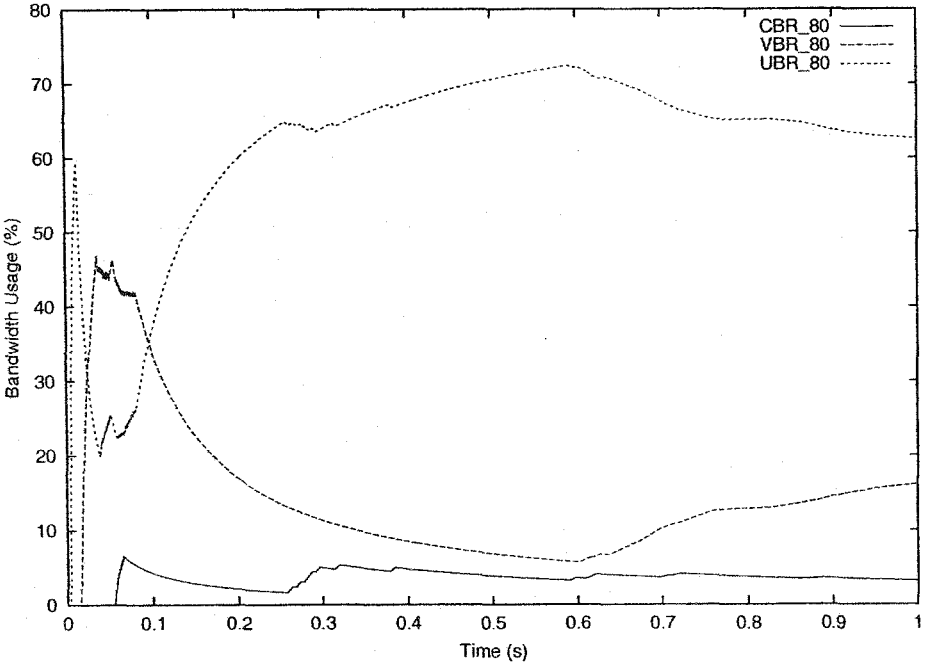


Figure 2.19: Burst response of DQRUMA, heavy load

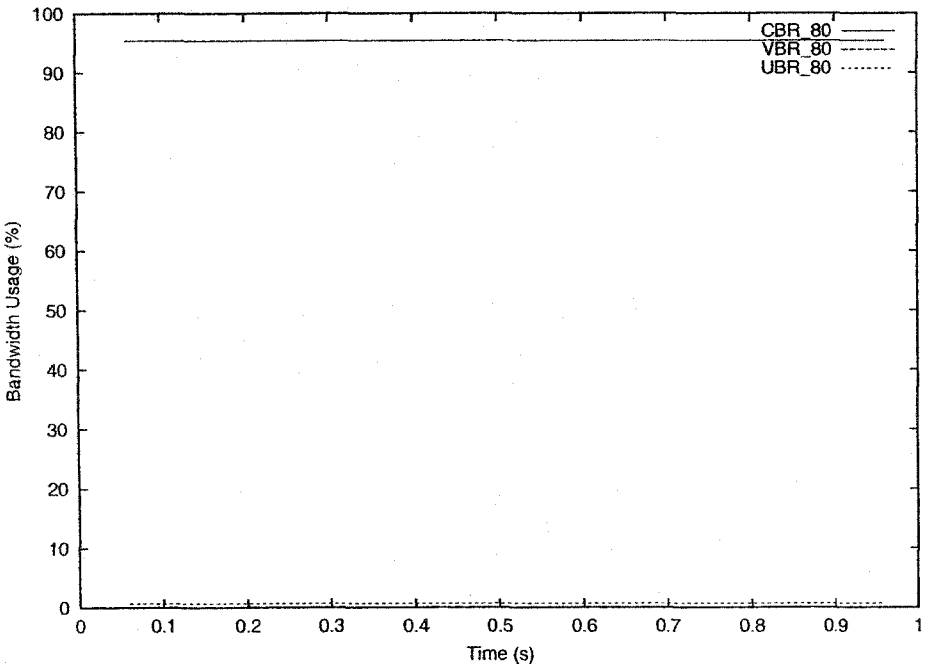


Figure 2.20: Burst response of DS-TDMA/CP, long talkspurt, heavy load

Chapter 3

BRICS

The second wireless multiple access scheme that we propose is based on CDMA. With this technology, every mobile user is assigned a pseudo-random binary sequence, also called a pseudo-noise (PN) code. Before transmission, the user data are *spread* (digitally superimposed) with its assigned PN code before used to key (modulate) the carried frequency of the transmitter. To correctly interpret the signal, the receiver must *correlate* (synchronize) its replica of the PN code used by the transmitter to the received signal and use exactly the same binary sequence to *despread* the data to their original shape. Transmissions from other users, although carried out in the same frequency band, are separable from each other owing to the *orthogonality* of their PN sequences. Thus, they only contribute to the level of white noise perceived by every single receiver. We call this disturbance the self-interference of the CDMA system.

The correlation between different multiple PN codes tends to fluctuate from bit to bit, and therefore multiple access interference (MAI) appears to be random in time and determined by the average interference level. The MAI is assumed to be Gaussian noise in some literature [84], but this assumption is usually false [85]. A particular code assignment and scheduled power control of transmissions may result in interference which, collectively, is not Gaussian. Since we only considered an one-cell system to study the protocol performance, we do not consider the MAI as Gaussian noise in this study.

The differences in propagation path loss between different mobile stations and the base present a problem called the *near-far* problem. Namely, the signals received by the base station from a mobile located close to the base station will be stronger than the signals received from another mobile located, e.g., at the cell boundary. Hence, distant users will be dominated by close users. To overcome the near-far problem, transmission power control is required as part of the medium access/bandwidth allocation scheme, such that the level

of received signals is kept roughly the same for all mobiles.

3.1 The protocol goals

To support multimedia applications with a CDMA scheme, we propose a time-slotted CDMA protocol, dubbed BRICS, which stands for *Base Rate Incremental Coded Service.* As two integral components of our solution, we present a quick code acquisition system and an air interface in which the uplink signaling channel is based on code-domain minislots. We also propose a method to explicitly exploit silent periods in voice activity. The protocol objectives are to quickly respond to the inherent variability in offered load (including voice and data traffic), and to recycle as much bandwidth as possible without compromising the integrity of the sessions in progress. As indicated by our performance studies, the proposed protocol efficiently accommodates multiple traffic classes with different bandwidth requirements and QoS expectations.

3.2 Protocol prerequisites

We consider a single-cell system with one base station (BS) and a variable number of mobile stations (MS) communicating exclusively with the base. Different mobile stations may be engaged in sessions of different types, e.g., voice, video, or data. Different types of sessions have different QoS requirements, which translate into different transmission rates and channel conditions, i.e., bit error rates. We assume that the system is frequency division duplex (FDD) on the the downlink (BS to MS) and the uplink (MS to BS). As in the case of DS-TDAM/CP, the handoff part of the overall communications system implementing our protocol is beyond the scope of our present study.

3.2.1 A multi-code transmitter for BRICS

The BRICS protocol is based on multiple code CDMA (MC-CDMA) [36, 42] and assumes the same prerequisites as WISPER [40], with the mobile transmitter structured, e.g., as shown in Figure 3.1. The carrier modulation is assumed to be binary PSK.

According to MC-CDMA, all transmissions are carried out at the fixed *basic* rate R_b. Using different codes, a single mobile can transmit up to M packets simultaneously, where M denotes the mobile's *maximum rate.* When a mobile needs m times the basic rate ($1 \leq m \leq M$), it converts its data stream, serial-to-parallel, into m basic rate streams, spreads each basic rate stream with a different code and superimposes them before up-

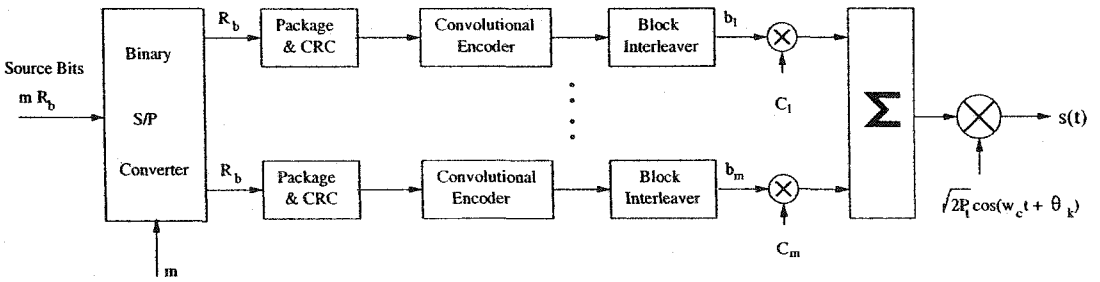


Figure 3.1: MC-CDMA transmitter

converting for radio transmission. In Figure 3.1, m denotes the rate at which the mobile is transmitting ($m = 1$ means that the mobile is transmitting at the basic rate R_b). This is also the serial-to-parallel conversion ratio and the number of active parallel branches of the transmitter. C_i, $(i = 1, \ldots, m)$, are the spreading codes that the mobile employs to implement the multiple *code channels* when transmitting at a rate higher than the basic rate.

The codes C_i are generated by a subcode concatenation scheme [36]. Each mobile admitted to the system is assigned a primary code. The primary codes of different mobiles are different pseudo-noise (PN) codes, which need not be completely orthogonal. If C_{PN} is the primary PN code of a mobile transmitting at rate m, the spreading codes, C_i, $(i = 1, \ldots, m)$, are derived from C_{PN} as

$$C_i = C_{PN} \times D_i, \quad D_i \perp D_j, \quad i \neq j \tag{3.1}$$

where D_i $(D_i \perp D_j, i \neq j)$ are from a set of orthogonal codes (e.g., Walsh codes), so that $C_i \perp C_j$, $i \neq j$, is guaranteed. Theoretically, this orthogonality is maintained at the receiver since the propagation variations on the parallel codes are the same. Multiple streams transmitted from the same mobile do not interfere with each other if they arrive at the base station simultaneously, but this may not be so in the face of multi-path fading. Thus, to be realistic, we do not consider the subcode concatenation advantage in reducing the interference.

The transmission power expedited by a mobile must increase along with the transmission rate m to provide the same signal-to-interference ratio (S/I) for each of the parallel code channels. The total transmission power is denoted by P_t in Figure 3.1. θ_k stands for the phase shift of the carrier frequency, which is usually zero.

3.2.2 A multi-code receiver for BRICS

In order to consistently decode the information transmitted over a BRICS channel, the receiver must demodulate and despread the received radio waves. As a prerequisite, the receiver must establish synchronization between the spreading pseudo-noise (PN) code on the received radio waves and their locally generated replicas. This operation is accomplished in two stages: coarse alignment (code acquisition) and fine synchronization (tracking). In the uplink, the code acquisition typically occurs before the carrier phase synchronization, and a non-coherent detection scheme is usually applied.

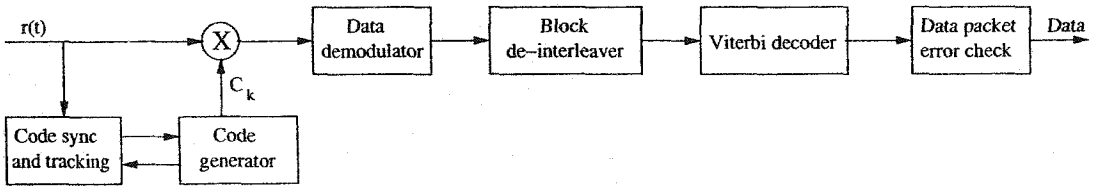


Figure 3.2: MC-CDMA receiver

As each code channel uses a different spreading code, it needs a separate correlator for code acquisition. Theoretically, if all m channels experience the same delay, they can all share a single code acquisition circuit. There are reasons, however, why it seems more reasonable to provide a separate circuit for each code channel—e.g., to take advantage of the multipath gain, as in a RAKE receiver, where multiple code acquisition circuits are needed for each code channel. The high level layout of a single path multi-code BRICS receiver is shown in Figure 3.2. Owing to the time limitation for the code synchronization within one minislot (see Section 3.3), a quick and reliable code acquisition scheme is critical for the correct operation of the proposed protocol.

Code acquisition techniques have been extensively studied in the literature. Following [73, 74, 75], Figure 3.3 shows the high-level layout of a fast parallel code acquisition system that we propose for BRICS, built around a matched filter and utilizing the maximum likelihood strategy.[1] To increase the reliability of code acquisition and reduce the chance of a costly *false alarm*, the code acquisition system has two function blocks: a searching block and a verification block. The search block consists of N parallel matched filters, as shown in Figure 3.4(a). A detailed description of such filters and correlators can be found in [76]. The second function is carried out by a verification correlator shown in Figure 3.4(b).

Fast code acquisition in the reverse link (i.e., at the base) is more critical because of

[1] In fact, this is not a true maximum likelihood strategy, since the tests are carried out on different observations of the same received signal.

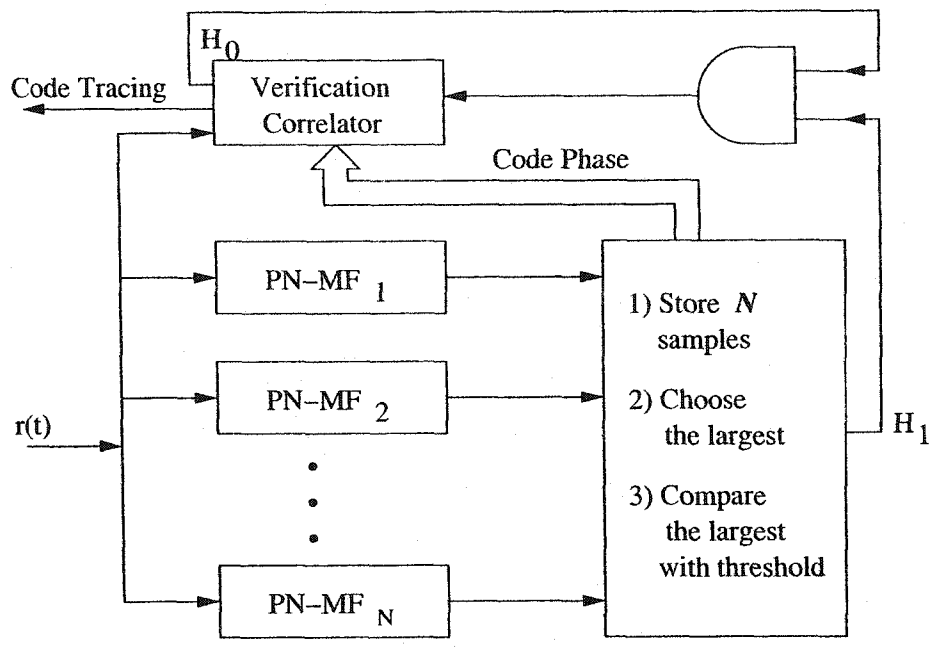


Figure 3.3: A PN code acquisition system based on parallel matched filters

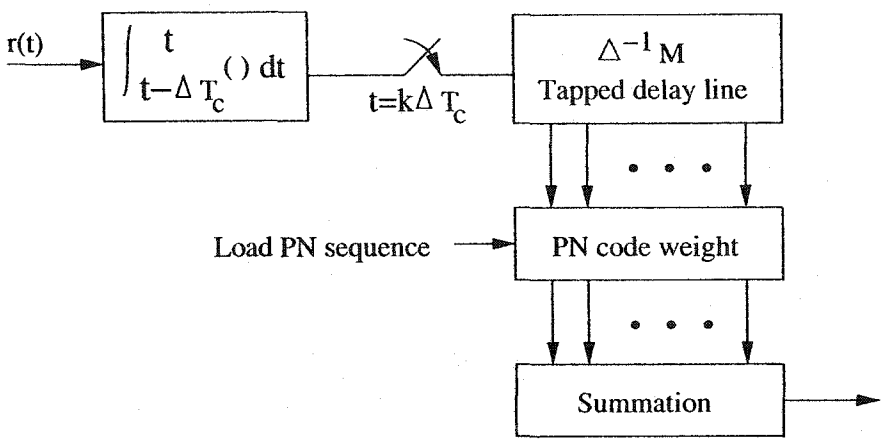
the compact organization of the uplink frame. To facilitate it, the mobile station transmits a fixed length unmodulated PN sequence (the acquisition preamble). We assume that the code is *long*, and partial-code correlation is applied in the matched filter. Since the uplink channel is slotted, transmissions are constrained to start at slot boundaries.

The mobile can calibrate its clock according to the time signal transmitted by the base station on the pilot channel. Therefore, the phase delay of the acquisition preamble at the base station is equal to the round trip time plus the signal processing time at the mobile station. Consequently, the uncertain region for code acquisition at the base station is determined by the maximum distance D between the base and the mobile plus the uncertainty δ of the signal processing time:

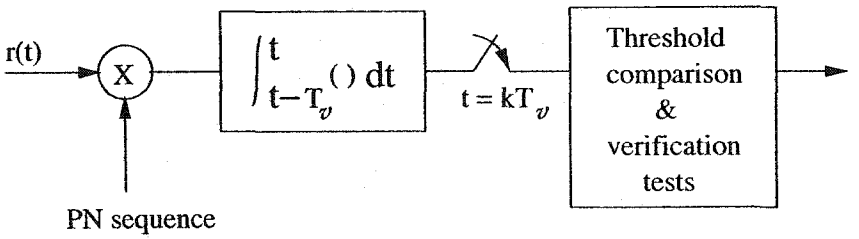
$$L = \frac{\frac{2D}{c} + \delta}{T_c}, \tag{3.2}$$

where L is the duration of the uncertain region expressed in the number of PN chips, c is the speed of light, and T_c is the PN chip duration. In a microcellular environment, D is small and so is L.

We assume that the code synchronization procedure starts at time zero (expressed in PN chips). Its first stage is code search. Within the search block, using N parallel passive non-coherent PN matched filters (PN-MFs), the uncertain region L is divided into N sub-sequences, each of length $M = L/N$. Each PN-MF is loaded with (matched to) one of the



(a) PN Code Matched Filter



(b) Verification Correlator

Figure 3.4: Matched filter and correlator in a PN code acquisition system

N subsequences. The number of taps (or search cells) on each delay line in the matched filter is M/Δ with the delay of ΔT_c between successive taps. A typical value for Δ is $1/2$. In MT_c seconds, NM/Δ cells are searched, with each cell corresponding to one of the possible MN/Δ phases in the uncertain region. The largest sample and the corresponding code phase from each of the N parallel PN-MFs are stored and compared. With only one sample being stored, this reduces the memory requirements and signal processing time, e.g., in comparison to [73].

If the largest sample among the N outputs of the filters exceeds a threshold γ_1, a tentative synchronization (designated as a *hit*) is assumed, and the corresponding phase is used to initiate the correlator and the verification process is started. The searching continues until a true *hit* is declared in the verification process, or the preamble runs out, whichever happens first. In the former case, code tracking is started; otherwise, the code synchronization is presumed lost.

During the searching process, every LT_c seconds, the N PN-MFs are reset with a new portion of the PN code shifted by LT_c seconds of delay. The matched filters must be

programmable to be able to load different code portions. The verification process lasts for a specified amount of time. Within that time, a number of independent correlation tests, say A, will be carried out. Each correlation test takes an integration time of $T_v = KT_c$ seconds, where K is the number of code chips integrated and tested against threshold γ_2—see below. The total verification test takes AT_v seconds.

If at least B out of A tests exceed the second threshold, γ_2, the code acquisition is assumed and the code tracking system takes over the code synchronization. Otherwise, a false alarm is declared. In the latter case, if a new tentative *hit* was found already, say at time T_h, the verification process is immediately restarted with the phase shift of $t - T_h$, where t is the current time. Otherwise, the verification is aborted and postponed until the next *hit*. The verification is also aborted and reset if within the current search interval of MT_c seconds a new sample is found that is bigger than the one that triggered the previous *hit*. After that interval, however, the correlator is never reset unless it detects a false alarm. All these functions, including cell sampling, comparison and verification are carried out essentially at the same time, possibly separated by a few clock cycles. If the clock is fast and the delay is contained within one PN chip interval, this delay can be ignored.

An analytical estimate of the acquisition probability for a single matched filter is given in [74]. Since we employ N parallel filters, which match to multiple segments of the PN code, the acquisition probability in our case is considerably higher than the estimate in [74], and of order $1 - (1 - P)^N$, where P is the probability for a single filter. This extrapolation may be slightly overoptimistic, however, because the N samples are not strictly independent (although correlations among them are not easy to capture).

On the other hand, one can suggest a few directions for improving the accuracy and increasing the speed of the code acquisition algorithm even further. As subsequent slots transmitted by the same mobile (using the same key) will tend to be located rather closely in time domain, the uncertainty interval for code acquisition can be centered around the exact phase found for the previous slot. Also, this interval can be set much tighter than that prescribed by Formula 3.2, based on how late the current slot follows the last one for which the code was successfully acquired. Even if the mobile moves at a high speed, it cannot move too far between two consecutive slots, e.g., transmitted in two consecutive frames. Consequently, the primary factor contributing to the uncertainty of the code phase in Formula 3.2, i.e., the distance between the mobile and the base, will tend to change very little between consecutive acquisitions. Notably, this will tend to be the case for high-rate sessions, for which the quality of the code acquisition algorithm is especially important—

from the viewpoint of the overall error rate and effective bandwidth. Only if the mobile has been silent for a very long time, may the uncertainty of the code phase reach its worst case estimate given by Formula 3.2.

Our approach based on multiple parallel filters may seem expensive at first sight. However, although it increases the hardware complexity of the acquisition circuits, it poses no fundamental problems other than pure complication. With the rapid progress in digital signal processing and VLSI technologies, previously impractical and expensive parallel schemes become quickly feasible [77]. Other solutions proposed in the literature, notably WISPER [40], also assume fast code acquisition within multiple slots of a single frame, and in this sense our proposal is no more costly to implement than those ideas.

3.3 BRICS: Protocol Description

One important feature of BRICS is that, in contrast to many other CDMA protocols proposed in the literature (e.g. [37, 45, 46]), its bandwidth allocation scheme is completely deterministic. This means that a mobile granted the right to transmit does it with probability 1 within a precisely described collection of slots, as opposed to using a persistence factor $p < 1$ as its transmission probability. This approach turns out to be more efficient in exploring the patterns of mobile activity, in particular, when voice must be integrated with other traffic types.

3.3.1 Logical channels

Since the performance of a mobile system is usually limited by the capacity and flexibility of the radio link from MS to BS (which must be multiplexed among multiple contending mobiles), only the structure of this link (the uplink) is discussed in detail. The downlink can use a similar structure, but access to it is very simple because the base station is the only transmitter on this channel.

The uplink is organized into time frames, with every frame partitioned into a number of logical channels that may span both dimensions, i.e., time and code. A channel is described by a time slot, i.e., its starting moment relative to the frame boundary and duration, its code, and the intended received power at the base station. For illustration, the frame can be envisioned as consisting of multiple layers of slots (resembling a brick wall, as in Figure 3.5) that coincide (or overlap) in time (the horizontal axis), but are separated by different codes (along the vertical axis). The amount of power assigned to a channel corresponds to the thickness (or height) of the corresponding brick.

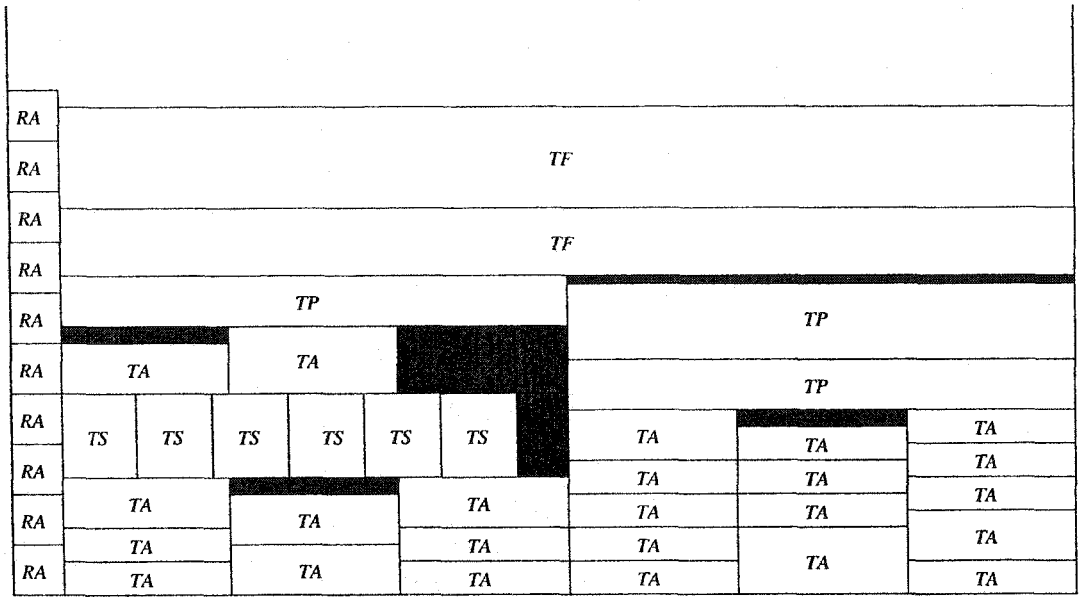


Figure 3.5: BRICS frame structure

Although every slot occurs at a definite time position within a frame (and its position along the horizontal dimension of the wall is precisely defined), the exact structure of layers is irrelevant. Owing to the orthogonality of the codes assigned to the overlapping slots, the exact value of the code (which would correspond to the exact position of the slot along the vertical axis) is not important, and, consequently, it makes little sense to say that a given slot belongs to a specific layer. What matters, is the total number of slots (codes) being assigned across a given time position within the frame, which, together with the combined amount of transmitter power allocated to these slots (represented by their thickness), determines the thickness of the entire wall at that location. By looking at Figure 3.5, one might get an impression that the problem of efficient slot allocation consists of trying to fill as many individual holes (the shaded areas) as possible, whereas in fact it boils down to minimizing the maximum height of the wall understood as the sum of all heights of bricks across a given location, ignoring the gaps. This maximum height determines the maximum amount of power at which the frame is received by the base station (see Section 3.3.4).

Except for the *RA* (random access) slots, which must all occur in the same time location (i.e., at the very beginning of the frame), the shape of the remaining slots may vary across the vertical (code) dimension. This is different, e.g., from the approach taken in [40], where different slots occurring at the same time must have identical properties. On the other hand, the *RA* slots play in our protocol a similar role to the contention slots in [40], i.e., they are used for registering bandwidth requests of the mobiles.

Following the *RA* slots, the remaining portion of the frame is built of four slot types. The granularity of bandwidth allocation is determined by the size of the standard (basic) slot denoted by *TA*. Every transmission slot is allocated in time on a *TA* boundary, starts with a code acquisition preamble (Section 3.2.2), and extends for the total duration of an entire number of *TA* slots. This alignment, besides simplifying code acquisition [78], reduces the amount of information needed to describe slot locations, which is important from the viewpoint of downlink signaling.

The duration of the *TA* slot is selected in such a way that a single voice session requires exactly one *TA* slot in its active phase. The second slot type, *TS*, is used to build signaling channels, needed by the mobiles admitted to the system to provide the base station with a feedback regarding their dynamic bandwidth requirements and received power. These slots are necessarily allocated from the beginning of the frame. They cannot appear too close to the end, because the base must be able to process the information contained in them before announcing the layout of the next frame. Since they are shorter than the *TA* slots, the last signaling slot may be followed by an unusable gap.

The third slot type is very simple: it spans the entire frame space and is intended for high-bandwidth sessions. We call it a *flat slot* (or a flat channel) and denote it by *TF*. To increase the flexibility of flat allocation, we admit *partial flat slots*, denoted by *TP*, spanning the width occupied by several *TA* slots, but less than the whole frame. Since the allocation of partial flat slots is considerably more complex than for the remaining slot types, it makes sense to impose (possibly quite drastic) restrictions on their size and/or position within the frame—to bring the complexity of the scheduling algorithm down to a manageable level. In our virtual implementation of the protocol (discussed in Section 3.4), we have assumed that a partial flat slot is half the size of the (full) flat slot, and that it must be aligned on a half-frame boundary.

By using slots with multiple sizes, the protocol reduces the amount of bandwidth wasted on framing, i.e., acquisition preambles and guard spacing between slots. Although in principle any bandwidth requirement could be fulfilled by a collection of slots of the same standard size, this approach, besides wasting bandwidth on slot boundaries, would complicate the scheduling algorithm at the base and significantly increase the length of signaling messages on the downlink channel.

To reduce the complexity of code allocation, the primary PN code for each mobile is assigned at the moment when the station is admitted to the system. If the mobile needs more codes to sustain a high rate session, it will create them as described in Section 3.2.1,

and the base station will use the same method to generate their replicas. Furthermore, a (small) predefined number of codes are reserved for uplink signaling, i.e., to be used for status transmission within the *TS* slots. The corresponding number of dedicated receivers at the base are responsible for listening to the signaling channels. As a single signaling channel occupies a single *TS* slot in the uplink frame, we will identify signaling channels with *TS* slots (and call them *TS* channels). When scheduling a mobile transmission, the base station will indicate to the mobile the time position of its slot, the slot type, and the parameters of the signaling (*TS*) channel. The signaling channels are given a higher error protection, i.e., more power, than other (typical) channels.

The *RA* channel can be viewed as a special case of a signaling channel. Depending on the number of dedicated receivers at the base, a number of PN codes, called the *RA-PN codes*, will be reserved for the *RA* channels. An *RA* channel is formed with a *RA-PN* code and a *RA* slot at the beginning of each frame. The number of *RA-PN* codes is equal to the number of *RA* channels. Given the time separation of the *RA* slots from other components of the frame, the risk of a packet error or loss caused by a contention from a new mobile is greatly reduced.

The indivisible time unit of bandwidth allocation within the frame is one *TA* slot. Although a *TS* slot is shorter than *TA*, the total length occupied by one "layer" of *TS* channels is determined with the granularity of *TA* (because shorter leftovers are unusable).

The type of the channel allocated to a session depends on its bandwidth requirements and on the current system load, i.e., bandwidth availability. For example, *TA*-based channels are well suited for isochronous on-off sessions, e.g., voice. Such sessions essentially operate in a TDMA fashion, as in most of the existing commercially successful digital cellular systems, e.g., GSM. The simple model of a voice session assumed in our experiments (Section 3.4.4) is a binary on-off process, with the burst rate coinciding with the rate of one *TA* slot per frame. If needed, the protocol can be easily extended toward accommodating voice sessions with discretized multiple levels of burst, e.g., as in IS-95, by allocating to them several *TA* slots per frame. In principle the same approach can be used for allocating other channels with diverse bandwidth. In particular, a single mobile might receive a number of (possibly non-adjacent) *TA* slots within the same frame, possibly spanning multiple codes, in a way corresponding to the allocation of a (partial) flat channel. However, such a solution would complicate the processing at the mobile station, increase the length of downlink signaling messages (which would need to convey information about multiple slot positions), and waste a considerable amount of uplink bandwidth on slot boundaries. Consequently, if useful at

all, this approach should be limited to very special and restricted scenarios, e.g., allocating no more than two or three *TA* slots per session.

Continuous high bandwidth sessions, e.g., video, are best handled using flat channels. If the rate of a single flat slot *TF* is insufficient to accommodate a session, the mobile can be assigned multiple flat slots, i.e., transmit at $m > 1$ times the basic rate R_b (Section 3.2.1). Low priority ABR/UBR type sessions can use whatever bandwidth is left over after the time-critical sessions have been accommodated. Such sessions are easy to handle by the bandwidth scheduler, because they can be used to "fill the holes" left over by the higher priority sessions, whose requirements are more stringent.

A similar approach to bandwidth allocation can be employed in the downlink channel. Since the base station enjoys exclusive access to the downlink channel, there is no need for the *RA* slots in this case. Another difference is in downlink signaling. Instead of using multiple signaling channels, the base station transmits a single *control packet* aligned at the end of the frame and occupying an equivalent of a (partial) flat channel. With this solution, no bandwidth is wasted on slot boundaries, and the control messages can be of variable length, e.g., depending on the number of slots/codes assigned to a given mobile. All mobiles tune in to the control packet (using a predefined code) and identify its relevant fragments (individual control messages) specifying the locations of their slots, the positions and keys of their uplink signaling channels, and the power adjustment. The number of predefined signaling codes is small, e.g., 2-4; so this part of the control message occupies a trivially small amount of space. With the power levels quantized into a reasonable number of discrete states (e.g., 256, 512, or 1024), the size of a single control message is going to be small and so is the total amount of bandwidth needed for downlink signaling (see Section 3.4 for a numerical example).

3.3.2 Medium access

When a mobile station wants to initiate a session, it randomly selects one of the *RA* channels and sends a request to the base station using a modified ALOHA protocol. Besides the preamble for code acquisition and carrier synchronization, the access request packet transmitted in the selected *RA* channel consists of the following components: mobile ID, service type, resource requirements, delivery deadline, and transmitted power level. The exact specification of resource requirements depends on the service type and may include transmission rate, message length, or be empty. For example, the characteristics of a voice session are always the same and completely determined by the service type. If the base

station correctly receives the access request packet, it will establish a connection with the mobile station (and assign it a dedicated signaling channel) as discussed in Section 3.3.3.

Of course, it is possible that an access request packet will not make it to the base station. For example, several mobiles may transmit within the same *RA* channel. Owing to the capture effect in CDMA systems, collisions need not be always damaging to all the involved parties, i.e., one of those mobiles may succeed; however, the remaining stations will fail to convey their requests to the base. Even if there is exactly one transmission within an *RA* channel, it may fail because of a high MAI level (heavy contention to other *RA* channels) or too low transmission power used by the mobile.

To improve the performance of the medium access scheme, failed requests are handled both by the mobiles as well as the base station. As part of the control packet broadcast by the base station to announce the layout of the forthcoming frame, the base indicates to the contending mobiles the current *contention permission status*, i.e., a sequence of binary flags indicating which traffic classes are allowed to compete for bandwidth. With this mechanism, similar to the *contention permission flags* in DS-TDMA/CP—Section 2.3.2), the base station is able to inhibit lower priority sessions when bandwidth becomes scarce. By thresholding the perceived noise level in the *RA* channels (possibly accounting for inter-cell interference), the base may also selectively restrict contention to high priority classes if that level appears to be too high.

On the mobile's end, we propose a combination of p-persistent behavior (which is a standard approach in ALOHA systems) with adjustments of transmitter power. Following an unsuccessful attempt, indicated by the lack of a response from the base station in the next downlink control message, the mobile station will reduce the probability of transmission p and, at the same time, increase the power for a subsequent attempt. In this way it will become less aggressive with its requests, while improving its chance for a successful reception by the base. Also, by using more power, mobile stations that have failed are given priority over new contenders: they are more likely to win considering a given level of MAI from other codes, and more likely to be captured if other contenders (transmitting at lower power) happen to pick the same code. This policy, i.e., favoring delayed losers over newcomers, has been demonstrated to improve the stability of collision based schemes [82].

With our generic scheme, the mobile issues its first attempt with probability $p_0 = 1$ using power $P_0 = P_a$, where P_a denotes the initial (starting) power level. Following a failure, the station executes $P_{i+1} = \min(P_i \times \delta_P, P_{max})$, where $\delta_P > 1$ is the power increment factor and P_{max} is the maximum power, and $p_{i+1} = \max(p_i \times \delta_p, p_{min})$, where $\delta_p < 1$ is the

probability decrement factor and $p_{min} > 0$ is the minimum probability of transmission. In Section 3.4.2, we recommend some specific values for these parameters.

One possible way to determine the value of δ_p is to take the transmitted power into account, that is, set $\delta_p = \frac{P_c - P_t}{\Delta_P}$, where P_c is the mobile's maximum transmission power, P_t is the transmission power that the station used in the last attempt, $\Delta_P = P_c - P_a$ is the adjustment range of the transmission power at the mobile. The value of δ_p produced by the above formula accounts for the access interference that the transmission will generate. The higher the transmission power, the higher the probability that the access request will be successfully received by the base station and also the higher the interference that the mobile station will incur at the base to other mobiles. The decrease of the transmission probability p is thus compensated by the higher likelihood of a successful reception by the base, while at the same time reducing the average interference level generated by the mobile while contending for access to the base. Additionally, the procedure of power adjustment carried out during the contention phase can yield a coarse initial power level for the fine power control needed after the connection has been established. We will not study this issue in more detail and assume that the required power level can be realized both at the base station and at the mobiles.

3.3.3 Sessions

All transmissions in the system are carried out within the frameworks of their sessions. This means that the operation of the cellular system is connection-oriented.

Before a mobile station can make a data transmission, a virtual connection must be established between the mobile and the base station. Such a connection constitutes a *session*, which spans the period from the moment when the base receives an access request from the mobile, until the mobile explicitly indicates that the session is over. At least two channels, a signal channel and a traffic channel, are needed for a session to remain active. Issues like handoffs and timeouts to detect failed mobiles are not covered by our study.

When the base station successfully receives an access request from a mobile station, it will immediately respond with a signaling channel assignment. The mobile station will inform the base through the signaling channel about the change in its bandwidth requirements, the quality of the downlink signal received from the base (represented by the E_b/N_0 ratio), and the uplink transmission power level. A session tear down request can be viewed as a special case of bandwidth requirement. In response to such a request, the base will terminate the session and release the resources reserved for this mobile.

Besides the signaling channel, the mobile station will also be assigned upon a session setup one or more traffic channels, depending on the mobile's requirements, session type, intermittent system load, and the available bandwidth. Only after the traffic channel has been explicitly assigned by the base, can the the mobile station start transmitting its data. The mobile station will adjust its resource requirements during the session according to its activity (its traffic flow status). If the mobile's data buffer drains, it can release some resources (request less bandwidth in a subsequent frame). If the mobile runs out of burst, all the traffic channel(s) can be released temporally by setting the required transmission rate in the signal packet to zero. When it becomes ready to transmit, the mobile can re-acquire the released traffic channel(s) via the signal packet, which is never released for as long as the session remains alive. The base station can preempt lower priority sessions to satisfy the dynamic requests of higher priority mobiles.

3.3.4 Bandwidth allocation and admission control

The responsibility for allocating bandwidth to the multiple mobiles within the cell, in accordance with the QoS expectations of their sessions, rests with the bandwidth scheduler at the base station. The primary criterion that tells apart different traffic classes is their priority. A bandwidth request received by the base is stored in the queue corresponding to the traffic class specified in the request. The multiple request queues are examined by the scheduler in the decreasing order of their priorities. Within every queue, the requests are arranged according to some class-specific criteria (e.g., delay bound, transmission rate) or FIFO. A different scheduling algorithm, specific to a given traffic class, is employed within each queue.

BRICS allows multiple packets belonging to different traffic classes to be transmitted at the same time. Every code assigned in the uplink frame requires a dedicated receiver at the base station "tuned" to that code. Thus, the number of mobiles that can be scheduled at the same time (i.e., within one time slot) also depends on the number of receivers available at the base.

BRICS schedules the transmission requests in a "brick wall" manner. The height of the "brick wall" filling a single uplink frame (Section 3.3.1) is determined by the admission control algorithm. Admission control criteria in interference-limited CDMA systems have been extensively discussed in the literature [70, 79, 80, 81]. To provide a satisfactory reliability of transmission for a given service, the system must ensure that the bit error rate (BER) does not exceed the service-specific maximum. This parameter, together with the

bandwidth specification, constitutes an important QoS criterion in a mobile environment. The BER specification can be mapped to the bit energy to noise spectral density ratio E_b/N_0 [70, 80].

Along these lines, the responsibility of the admission control part of our scheduler is to minimize the total transmitted power from all the mobiles and satisfy the minimum required E_b/N_0 for every session. Since, in contrast to [80], all the allocated channels in BRICS are always active, by extending the the results from [80] and setting the active factor to 1, we obtain the following admission constraint:

$$\sum_{k=1}^{K} \alpha_k < 1, \qquad \alpha_k = \frac{(E_b/N_0)_k}{W/R_b + (E_b/N_0)_k} \tag{3.3}$$

where K is the number of simultaneous code channels in the time slot, $(E_b/N_0)_k$ is the E_b/N_0 requirement for k-th code channel, W is the total spread bandwidth and R_b is the base channel rate (W/R_b is generally referred to as the spreading gain).

Under the constraint of Formula 3.3, the minimum power assignment is prescribed by

$$P_i = \alpha_i(\eta + P), \qquad i = 1, \ldots, K \tag{3.4}$$

where η is the background noise and P is the minimum total received power given by

$$P = \left(\sum_{k=1}^{K} \alpha_k \eta\right) \Big/ \left(1 - \sum_{k=1}^{K} \alpha_k\right) \tag{3.5}$$

While building the structure of the next uplink frame, the base station keeps an allocation table indexed by time slots (located at the time boundaries of *TA* channels—see Figure 3.5). Each entry in that table includes the current "height" of the "brick wall" across the slot and the list of requests accommodated at that location. With the criterion given by equation 3.3, the height of the wall at each slot location is the sum of all α_k falling in the slot. Suppose that *TA* and *TF* are the only channel types allocated by the bandwidth scheduler. A new *TA* channel is allocated at the location with the smallest height, which results in the update of a single entry in the allocation table, while the addition of a new *TF* channel simply raises the height of the entire wall (the values of all entries) by the same amount.

Formally, as different *TA* channels may have different values of α_k (and contribute different amounts to the local height of the brick wall), the optimal assignment of those channels to the frame is NP-hard (being trivially equivalent to bin packing). However, the maximum error (viewed as the deviation from the minimum height) resulting from the

simple greedy approach is bounded by the maximum height of a single brick, i.e., $\max_k \alpha_k$. Consequently, instead of searching for better approximations and more complex algorithms that would necessarily inflate the computational cost of the bandwidth scheduler, it makes better sense to simply assume that inefficiency of this magnitude is acceptable.

The addition of partial flat channels complicates the situation a bit further. Consequently, we postulate that high priority sessions be restricted to basic (TA) and full flat (TF) channels, whose allocation is straightforward and poses no problems. Following this "rigid" allocation phase, the leftover frame space can be partitioned among the more flexible lower priority sessions. In this way, instead of trying to solve a computationally difficult problem of partitioning the remaining chunks of bandwidth into a number of rigid chunks, we reverse the problem and allocate whatever chunks come out handy to sessions that know how to handle them.

The granularity of bandwidth allocation within the uplink frame is one TA slot. The load of a TA slot is defined as the height of the brick wall across the code dimension of that slot, or, more formally, the total received power at the base station assigned to all TA slots occurring at the same time. For the purpose of allocating channels covering multiple TA slots along the time dimension, we define a *slot group* as any sequence of time-consecutive TA slots. The load of a slot group is defined as the maximum load of a TA slot within the group.

The length of a slot group, expressed as the number of TA slots along the time dimension, may correspond to an allocatable amount of bandwidth. An implementation of BRICS may impose restrictions as to the standard collection of "group lengths," i.e., allocatable chunks of bandwidth within a single layer of the brick wall (a single code). Whenever a channel of a given bandwidth is needed, it is allocated as a group of slots of the required length with the minimum load. This simple approach ensures reasonable load balancing without incurring too much computational cost at the base. To this end, the bandwidth scheduler sorts the slot groups according to their loads and updates those lists as it allocates bandwidth to the sessions in progress.

In summary, the objectives of the scheduling algorithm can be stressed in the following points:

1. Meeting the QoS requirements of the mobile sessions. To accomplish this, the scheduler processes the request queues according to their priorities, using class-specific criteria within every queue. For example, deadline-critical requests are processed in the order of increasing deadlines. The relative amount of power assigned to every

request is determined by Formula 3.3.

2. Making efficient use of the overall bandwidth. To this end, the scheduler attempts to balance the allocation of channels trying to maintain an approximately equal value of $\sum_{k=1}^{K} \alpha_k$ across the entire frame. Intentionally, under heavy load, this sum should be close to 1.

3. Generating as little downlink interference as possible. This means that the total received power is kept close to the minimum prescribed by Formula 3.5, with the individual power levels at the mobiles determined by Formula 3.4.

The bandwidth scheduler operates in cycles. Every cycle starts with the reception of all *RA* slots through which new mobiles register their sessions with the base. These new requests are appended at the end of the respective queues. Then the status of the sessions in progress is updated based on the received contents of the signaling channels *TS*. In particular, some sessions may be torn down and their descriptions removed from the queues. Having received the last signaling slot, the base is ready to process the request queues, allocate bandwidth, and announce the layout of the next frame. The announcement message is built and transmitted on the fly.

3.4 A sample configuration of the protocol

In this section we discuss a sample virtual implementation of BRICS whose purpose is to illustrate the ideas presented in the preceding sections (by using concrete algorithms and specific numerical parameters) and to provide a model for performance studies, whose results are discussed in Section 3.5.

We assume that our cellular system offers four types of service (listed in the decreasing order of priority): voice, video conferencing, file transfer, and SMS (short message service). The mobile stations access these services through the base station connected to a broadband wired network, e.g., ATM.

3.4.1 The radio channels

The assumed basic rate R_b of our radio channel is 500 kbps and the transmission rate needed to sustain a voice session in its active phase is 22 kbps. The total length of a single uplink frame is determined as $l_f = t_g + p + l_{ra} + N_{ta} \times (t_g + p + l_{ta})$, where t_g is the guard time (8 bits = 16 μs), p is the acquisition preamble length (32 bits or 64 μs), l_{ra} is the length of

the contention slot RA (96 bits or 192 μs), l_{ta} is the length of the basic slot TA (384 bits or 768 μs–ATM payload size), and N_{ta} is the number of TA slots in a single layer of the frame (20). All these numbers add up to $l_f = 8616$ bits or 17.24 ms, which is the back to back duration of a single uplink frame. The payload length of a single flat slot TF is determined as $l_{tf} = N_{ta} \times (t_g + p + l_{ta}) - t_g - p$, which yields 8440 bits.

The total amount of bandwidth available within a frame is determined by the basic rate R_b, the total spread bandwidth of the channel W, and the E_b/N_0 requirements of the individual sessions. In our model, we assume $W = 20$ MHz as the target bandwidth, although we consider some other values for comparison.

3.4.2 Contention resolution

Contention to the RA channels is resolved according to the generic strategy described in Section 3.3.2, using 10 codes, with $P_a = 80$ mW, $P_{max} = 200$ mW (this is also the maximum transmitter power of every mobile in the system), $\delta_P = 1$ dB, $\delta_p = 0.25$, and $p_{min} = 0.1$. These values imply three levels of persistence $(1, 0.25, 0.1)$ and five power levels. As determined by simulation, they result in the effective throughput of $3.0 - 4.0$ new requests per frame, assuming $E_b/N_0 = 6$ dB and some contribution from the capture effect.

3.4.3 Signaling channels

The total length occupied by a single signaling slot TS (including the guard and preamble) is 72 bits, with the payload restricted to 32 bits. These bits are partitioned into a 6-bit bandwidth specification (e.g., 2 bits for the channel type and 4 bits for the number of channels), 10 bits for received power indication, and 8 bits left for extensions.

Note that once a session has been established, deadlines for individual packets (which are the same for all packets in the session) need not be specified in the signaling channel. As the mobile keeps the signaling channel in every frame, the base knows when a bandwidth request belonging to a session in progress was issued, so it also knows how to calculate deadlines for individual packets. Should a signaling packet be lost, the mobile can indicate in a subsequent signaling slot the lag of the request expressed in frames, which information can be comfortably fit, e.g., into 4 bits. Note that the assumed length of the contention slot RA is 12 bytes, which has enough room for a detailed description of the requested connection type.

All signaling slots occupy up to two layers (codes) of the first half of the frame space. The number of signaling channels per one layer (code) is limited by 58, and the maximum

total number of signaling channels is 116, which is also the maximum number of different sessions that can be accommodated in a single frame. Logically, all the signaling channels are viewed as a single array indexed from 0 to 115. The coordinates of a signaling channel are completely described by this index, which occupies a single byte in a downlink control message (Section 3.3.1). A base station uses the least significant bit of this number to determine one of the two codes to be used for sending the signaling packet and the upper seven bits indicate the time location of the signaling slot within the frame with the granularity of 72 bits. The bit energy to noise density ratio (E_b/N_0) for signaling channels is 6 dB.

Whenever a new signaling channel is required (for a new session), it is chosen as the next available *TS* slot (the one with the lowest index number) that satisfies the admission condition given by equation 3.3. Active signaling channels are easily compressible into a contiguous range of *TS* slots covering the minimum possible range of *TA* slots, without unnecessary fragmentation. This is because the exact location of the signaling channel for every active mobile can be specified independently for every uplink frame. In this way the *TS* slots are always as close to the beginning of the frame as possible, while the number of available *TA* slots is also maximized.

The scheduling algorithm

Variables:

P_a The index of the next available signaling channel.

P_e The largest index of the active signaling channels. P_e is set to 0 when the system starts up or is reset.

Algorithm 1: scheduling signaling channels

1. Find the next available signal channel index starting from 1. Let P_a point to this signal channel.

2. For all successfully received access requests on *RA* channels perform steps 3 through 6.

3. Assign the next available signal channel, P_a, to the requesting mobile station and mark this signal channel as active.

4. If P_a falls into a new *TA* slot, update the load of this *TA* slot. Note that all *TS* channels are assigned the same received power; thus, this update is done only once per *TA* slot.

5. If $P_a > P_e$, set $P_e := P_a$.

6. Advance P_a to point to the next available signaling channel.

7. Rearrange the signaling channels.

3.4.4 The voice traffic

As for DS-TDMA/CP discussed in Chapter 2, the voice traffic in our model of BRICS is represented with the "On-Off" process described in section A.3.1.

3.4.4.1 Session structure

Table 3.1 lists the relevant numerical parameters of our voice model. A BRICS mobile views a voice source as being in one of four states: *Idle* (I), *Request* (R), *Talking* (T), and *Silent* (S). A voice source in BRICS has the same state transition as it in DS-TDMA/CP (see Figure 2.3). When a user initiates a call, the mobile station transits from *Idle* to *Request* and issues a connection request to the base station. The mobile remains in the *Request* state until the request is granted, i.e., the mobile is assigned a signaling channel *TS* and a traffic channel *TA* by the base station, or until its access request due date expires. In the latter case, the mobile will become *Idle* again and the user will receive a busy signal.

When the request is granted, a voice session is set up between the mobile and the base station. The mobile will be generating talkspurts interleaved with silence periods for as long as the session is not torn down. During the session, the mobile will inform the base about the intermittent state of its session (silent/talkspurt) via the signaling channel. When the session enters the silent state, its traffic channel will be temporarily released, but the connection will be sustained through the signaling channel. When the session gets back to the talkspurt state, its traffic channel will be reassigned in the next frame. Similarly, when the call is completed, the mobile will indicate this fact via a pertinent signaling message. The resources used by the connection will then be released by the base station and its

source rate	22	kbps
mean call duration	3	minutes
mean talkspurt length	1	second
mean silence length	1.35	seconds
deadline	10	seconds
threshold for disabling voice requests	10	%
QoS	5	dB

Table 3.1: Parameters of the voice model

description will be removed from the scheduler queue.

3.4.4.2 Bandwidth allocation

At the base station, new voice requests (arriving in RA channels) are queued at the end of the voice queue in their received order. Active requests, controlled via signaling channels, retain their positions in the queue until they are terminated and removed. The voice queue is the first queue processed by the bandwidth scheduler, which means that voice service receives the highest priority.

The bandwidth temporarily released by a voice session that has entered the silent phase can be assigned to lower priority sessions, but it cannot be used to accommodate a new voice session. In this way, the bandwidth scheduler makes sure that all admitted voice sessions can always be accommodated in their active states, even if they all happen to enter this state at the same time. We assume that in a modern mobile network, especially in a microcellular environment, voice traffic (using relatively moderate bandwidth) will constitute a relatively small proportion of the overall load. Consequently, instead of relying on statistical properties of a large number of voice sessions, and admitting more sessions that can possibly be active at the same time, it makes better sense to restrict their number and recycle their intermittent silent periods in lower priority high-bandwidth sessions.

The access contention permission flag for voice traffic (Section 3.3.2) is cleared if the population of unserviced voice requests exceeds a certain percentage W_0 of all requests in the voice queue (set to 10% in our model). This is intended to reduce contention under heavy load. Of course, if a pending voice request does not receive a traffic channel after its deadline, the call will be blocked and the request will be dropped from the queue.

Scheduling voice sessions is relatively simple because the amount of bandwidth assigned to an active voice session is always the same and equal to one basic slot TA at $E_b/N_0 = 5\,\mathrm{dB}$. The allocation algorithm tries to put a new voice slot into the lowest layer in the brick wall. The base station also keeps track of the total amount of bandwidth assigned to all voice sessions, including the ones that have temporarily become silent. Once all sessions in progress have been processed, new voice requests are considered in the order of their arrival and admitted if the total amount of bandwidth allocated to all voice sessions (assuming they are all active at the same time) does not exceed the frame capacity. If, following this operation, the percentage of pending voice requests exceeds the threshold W_0, the contention permission flag for voice requests is cleared.

3.4.4.3 The scheduling algorithm

We assume that at any stage during the execution of the following algorithm we know the lowest and highest loads over all *TA* slots within the constructed frame. While the present algorithm only needs to know the location of the least loaded *TA* slot, the most loaded *TA* slot determines the maximum load of the entire frame, which information is needed for allocating flat and partial flat channels, as well as for power assignment.

Constants and variables:

W_0 The threshold for disabling access contention of the voice service.

w_0 The percentage of unserviced voice access requests in the total number of requests.

V_s The *TA* slot with the lowest load.

P_s The *TA* slot with the highest load.

Algorithm 2: *TA* channel allocation

1. Test the admission condition 3.3 on slot V_s.
2. If the admission condition is not met, return allocation failure. Otherwise perform the steps 3 through 5.
3. Allocate slot V_s to the session (as the *TA* channel), and update the load for the slot V_s.
4. Update V_s and P_s to represent the slots with the lowest and highest loads respectively.
5. Return allocation success.

Algorithm 3: scheduling voice service

1. Temporarily reserve bandwidth for silent voice sessions. Logically, this can be accomplished by carrying out essentially the same allocation procedure as for active sessions, except that the *TA* slots "reserved" this way will be in fact available to the remaining traffic classes except voice (Section 3.4.5.2).
2. For each new voice service request in its request queue perform steps 3 and 4.
3. Call algorithm 2 to allocate a standard *TA* channel.
4. If the allocation fails, stop scheduling voice service requests and skip to step 6. Otherwise, move to step 3 for the next request.

5. If all the voice service requests are satisfied, set the voice contention permission flag and go to step 7.

6. Calculate w_0. If $w_0 > W_0$, clear the voice contention permission flag.

7. Release the bandwidth reserved for the silent voice sessions, i.e., mark of the *TA* slots reserved in step 1 as empty. Stop here.

3.4.5 The video traffic

We assume that this traffic type represents the teleconferencing video service, whose pattern is described by the DAR(1) (Discrete Autoregressive) model described in Section A.3.2.

3.4.5.1 Session structure

The numerical parameters of our video model are listed in Table 3.2. A video source can be in one of three states (see Figure 3.6): *Idle* (I), *Request* (R), and *Scheduled* (S). When a video session is started, the mobile transits from state *Idle* to *Request* and issues a bandwidth access request to the base station. When the request is granted, the station moves to *Scheduled* and starts transmitting packets in the next frame. If the mobile times out in state *Request*, it will return to the *Idle* state.

As implemented at the mobile's end, the session is buffered with the lag of 4 frames (or 69 ms), which allows the station to better tailor the current rate to the size of the *TF* slot and be more flexible with the bandwidth received from the base. As an admitted video session is guaranteed a minimum bandwidth in every frame, no per-packet deadlines need to be specified in uplink signaling messages.

The bandwidth needed by the session may fluctuate (even momentarily dropping to zero), but the session remains established (and keeps its signaling channel) until it is explicitly torn down by the mobile. Using the signaling channel, the mobile conveys to the base its dynamic bandwidth adjustments.

mean source rate	128	kbps
DAR(1) variance	5536	
DAR(1) correlation	0.98	
mean call duration	30	minutes
call deadline	10	seconds
threshold for disabling video requests	10	%
QoS	4	dB

Table 3.2: Parameters of the video model

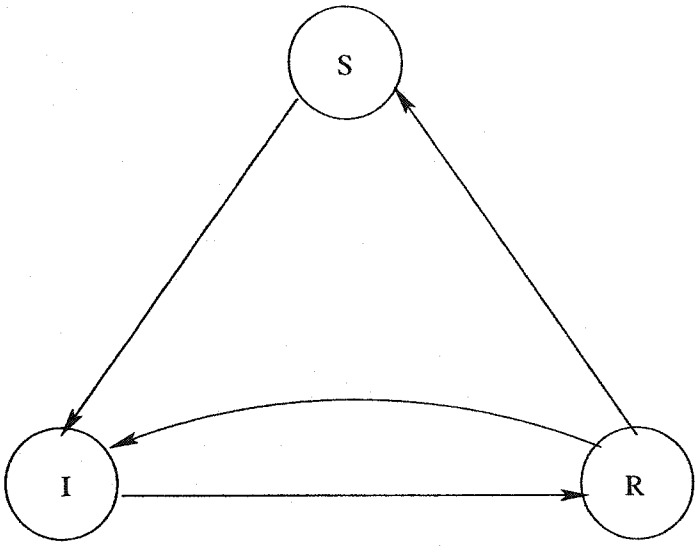


Figure 3.6: The video model

3.4.5.2 Bandwidth allocation

Because of its relatively high bandwidth requirements, a video session always receives an entire number of full flat channels TF. Similar to voice sessions, the video queue at the base consists of two parts: the requests in progress (already admitted to the system), and the pending requests waiting for admission. The requests in each list are stored in the increasing order of their due dates.

Every admitted video session is guaranteed a certain minimum amount of bandwidth b_{min}. Any bandwidth requested in excess of the minimum is dubbed *extra bandwidth* and scheduled frame by frame in a fair manner using the *equal degradation approach.* Let B_a denote the total maximum bandwidth available for teleconferencing service, after the higher-priority voice sessions have been accounted for, and B_r be the total extra bandwidth requested by the admitted video sessions. We define the current *service grade* as

$$G = \min(1, B_a/B_r) \tag{3.6}$$

The amount of extra bandwidth assigned to a video mobile is equal to Gb_r, where b_r is the extra bandwidth actually requested by the station.

While considering admitting a new video session, the bandwidth scheduler assumes that the session can start with the minimum bandwidth b_{min}, and admits the session if that much bandwidth is available after accounting for all video and voice sessions in progress. For this calculation, all active video sessions are counted with their minimum bandwidth

b_{min} and all voice sessions are assumed to be active (to make sure that an admitted video session will always receive some service).

The contention permission for teleconference service is held temporarily when the percentage of unserviced teleconference service requests is over the limit, and it is granted again when there are no pending teleconference service requests. In our implementation, this threshold is set to 10% of the video service request queue.

As video teleconferencing has the biggest bandwidth requirements among the considered traffic classes, it may make sense to impose a limit on the total number of video sessions being admitted at any given time. In our model, there is an option whereby the base station may deny admission to a new teleconferencing request if the number of active video session has reached a predefined limit, regardless of how much bandwidth remains available in the network.

3.4.5.3 The scheduling algorithm

Constants and variables:

W_1	The threshold for disabling contention for the teleconferencing service.
w_1	The percentage of unserviced teleconference access requests in the total number of requests.
G	The service degree as defined by equation 3.6.
P_s	The TA slot with the highest load.

Algorithm 4: scheduling video teleconference traffic

1. Assign the active teleconference sessions the bandwidth that they request, if the requested bandwidth is less than the minimum bandwidth b_{min}. Otherwise, assign the minimum bandwidth b_{min}.

2. Calculate the service degree G.

3. Assign to the active teleconference sessions the extra bandwidth based on the service degree.

4. Temporarily reserve bandwidth for active voice mobiles in the silent state and active video teleconference mobiles that do not use all of their minimum reserved bandwidth.

5. For each new teleconference service request in the request queue perform steps 6 through 9.

6. Test the limit on the number of admitted video teleconference sessions (if applicable), and test the admission condition, as prescribed by equation 3.3, on the P_s slot.

7. If the number of admitted video session has reached the limit, proceed at step 11. If the admission condition is not met, goto step 9.

8. Assign to the mobile station the requested number of TF channels, and update the load for each slot in the frame.

9. If there are more requests, continue at step 6.

10. If all the teleconference service requests have been satisfied, set the video contention permission flag and proceed to step 12.

11. Calculate w_1. If $w_1 > W_1$, clear the teleconference contention permission flag.

12. Release the bandwidth reserved for the voice mobiles in silent state and the unused bandwidth reserved for the video mobiles that do not use all of their minimum reserved bandwidth. Stop here.

3.4.6 The file transfers

We assume that there is no inherent delay requirement for a file transfer[2] and that this kind of session can use any available bandwidth and transmit at any rate physically available to the mobile. Thus, the bandwidth requirement specification in a file transfer request refers to the maximum required bandwidth rather than a strictly needed (or even preferred) amount. Our primary objective in handling file transfers is to maintain a reasonable degree of fairness among multiple sessions and avoiding unnecessarily splitting a single transfer into too many small slots.

The base station assumes, however, that there is a timeout after which a stalled file transfer session (that has made no progress because of the lack of bandwidth) is aborted and canceled. This is required to avoid blocking the signaling channel that may be needed to accommodate higher priority sessions.

3.4.6.1 Session structure

The session state diagram of a file transfer is shown in Figure 3.7. The mobile station can be in one of four states: *Idle* (I), *Request* (R), *Scheduled* (S), and *Waiting* (W). When there

[2]Deadlines in such sessions are usually subjective and determined by the patience of their users.

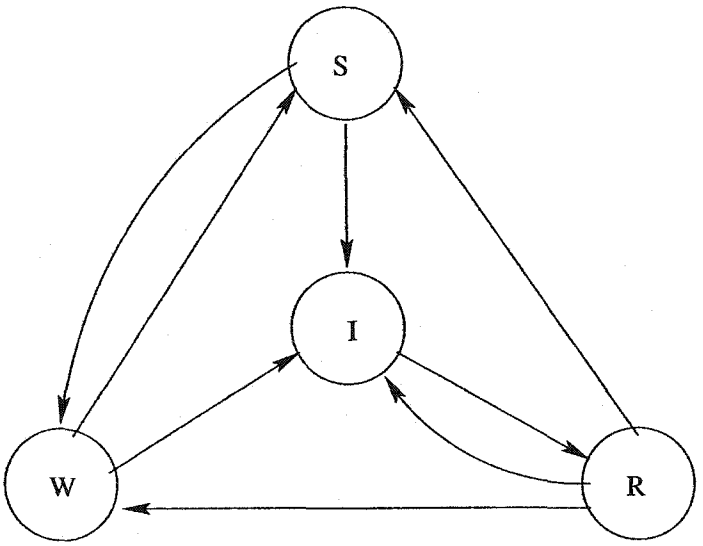


Figure 3.7: The file transfer model

is a file to be transmitted, the mobile station transits from *Idle* to *Request* and issues an access request specifying the maximum amount of bandwidth that the station could possibly use within one frame (which is the minimum of the file size and the rate capabilities of the mobile). The mobile remains in the *Request* state until it is either assigned a signaling channel, or it times out and is moved back to the *Idle* state. As soon as the mobile is granted some bandwidth, it transits to *Scheduled.* From now on, the station will keep sending via the signaling channel the updated bandwidth requirements, always requesting the maximum that the station could possibly use to expedite the remaining portion of the file. If the mobile station's request is accepted by the base, and the mobile receives a signaling channel, but no traffic channel, it transits from *Request* to *Waiting.* If the session gets aborted in this state on a timeout, the station will transit to state *Idle*; otherwise, it will receive a traffic channel and transit to state *Scheduled.* After each transmission in the scheduled data channels, the mobile station will stay in state *Scheduled* if it is assigned data channels in the next frame; otherwise, it will enter the *Waiting* state. If it times out in the *Waiting* state, the mobile transits to *Idle.* When the file transmission is finished, the station moves from *Scheduled* to *Idle.*

In our model, file transfers occur in bursts, with the burst duration and inter-burst intervals being exponentially distributed. During a burst, new files for transmission are generated at exponentially distributed intervals. The length of every file is exponentially distributed as well. Table 3.3 lists the numerical parameters of this process used in our

mean burst duration	30	minutes
mean inter-burst interval	30	minutes
mean interval between files in a burst	36	seconds
mean file length	104	KB
stalled session timeout	10	seconds
threshold for disabling file transfer requests	10	%
QoS	6	dB

Table 3.3: Parameters of the file transfer model

simulation experiments.

3.4.6.2 Bandwidth allocation

The bandwidth for a file transfer is allocated with the *preferred granularity* of one *TF* slot, in a Round-Robin fashion, with new requests being inserted at the end of the service queue, i.e., immediately behind the current position of the "next" pointer. If a mobile needs less than the full *TF* slot to complete its request, it is assigned a partial flat slot *TP*, i.e., 1/2 of *TF*, or/and a few basic slots *TA*. If all mobiles have been processed this way and some bandwidth still remains available, the procedure continues until all bandwidth has been assigned or all mobiles in this traffic class have been satisfied.

If no more *TF* channels can be allocated, because of the total height limitation of the brick wall, but *TP* channels are still available, the preferred granularity is downgraded to *TP*. If no *TP* channel can be found, but some basic *TA* channels are still usable, the preferred granularity gets down to *TA*.

To carry out the scheduling procedure described above, the bandwidth scheduler must keep track of the maximum height of the brick wall and the time position of the lowest *TA* slot, as for scheduling video and voice. The only increased complexity is the need to allocate partial flat channels *TP* aligned at half-frame boundaries. For this purpose, the scheduler stores and updates two additional values, namely the maximum height of the brick wall in the left and right half of the frame. A *TP* channel is allocated in the half with the lesser maximum height, and if both halves have the same maximum height, the right half is used first.

3.4.6.3 The scheduling algorithm

Constants and variables:

W_2 The threshold for disabling access contention of the file transfer service.

w_2 The percentage of unserviced file transfer requests in the total number of requests.

P_s The *TA* slot with the highest load.

L_0 The *TA* slot with the largest load among the first half of the standard slots within the frame. These slots constitute group 0 (see Section 3.3.4).

L_1 The *TA* slot with the largest load among the second half of the standard slots within a frame. These slots constitute group 1.

Algorithm 5: allocating a partial flat channel

1. If the load of L_0 is less than the load of L_1, select group 0 and test the admission condition (equation 3.3) on slot L_0. Otherwise, select group 1 and test the admission condition on slot L_1.
2. If the admission condition is not met, return allocation failure. Otherwise perform steps 3 and 4.
3. Assign to the mobile a partial flat channel TP covering the selected slot group and update the slot load for all slots within that group.
4. Return allocation success.

Algorithm 6: allocating a (full) flat channel

1. Test the admission condition prescribed by equation 3.3 on slot P_s.
2. If the admission condition is not met, return allocation failure. Otherwise perform the steps 3 through 4.
3. Allocate a TF channel (covering the entire frame) and update the load of every slot in the frame.
4. Return allocation success.

3.4.6.4 Algorithm 7: scheduling file transfers

1. Process file transfer requests in a round-robin fashion performing steps 2 through 6.
2. If the mobile requests a standard slotted channel *TA*, invoke algorithm 2 to allocate a *TA* channel. Continue at step 6.
3. If the mobile requests a partial flat channel TP, invoke algorithm 5. If algorithm 5 succeeds, continue at 6. Otherwise, change the mobile's demand to a standard channel *TA* and proceed to step 2.

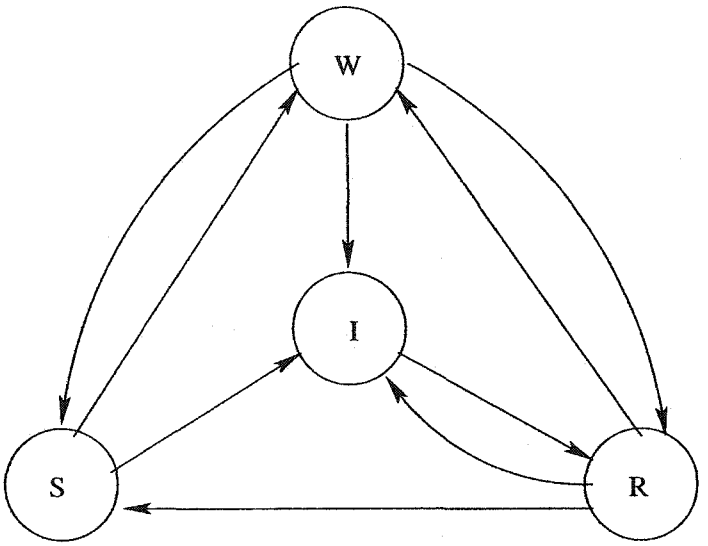


Figure 3.8: The SMS model

4. If the mobile requests one or more flat channels *TF*, invoke algorithm 6. If algorithm 6 succeeds, continue at 6. Otherwise, calculate the number of available flat channels *TF*.

5. If the number of available flat channels is not zero, change the mobile's demand to that number of available flat channels and continue at step 4. Otherwise, change the mobile's bandwidth demand to a partial flat channel *TP* and proceed to 3.

6. If there remain pending file transfer requests and the last request was satisfied, goto step 2.

7. If all the file transfer service requests have been satisfied (i.e., allocated some bandwidth), set the file transfer contention flag and stop here.

8. Calculate w_2. If $w_2 > W_2$, clear the file transfer contention permission flag.

3.4.7 The short message service

Short message service allows a mobile user to send or receive a limited number of characters of textual information. Conceptually, SMS service resembles electronic mail in that the user need not witness the actual transfer operation, which is happening in the background.

An SMS session involves two types of activities: one performed in the foreground and the other in the background. In the foreground, the mobile receives the message for transmission

mean message length	6250	bytes
mean interval time	11	seconds
deadline	2	hours
threshold for disabling SMS requests	20	%
QoS	6	dB

Table 3.4: Parameters of the SMS model

and stores it in a buffer. Essentially, this process consists of a single state: it sleeps waiting for a message arrival event, then it deposits the message in the buffer, and continues waiting until the next arrival event.

The other (background) process can be in one of four states (see Figure 3.8): *Idle* (I), *Request* (R), *Scheduled* (S), and *Waiting* (W). When there are messages in the buffer, the background process transits from *Idle* to *Request* and issues a connection request to the base station. The mobile remains in the *Request* state until any of the following events happens.

1. The mobile is assigned a signaling channel S_i and a traffic channel. In such a case, the mobile enters the *Scheduled* state.

2. The mobile is assigned a signaling channel S_i but not a traffic channel. In this case, the process enters the *Waiting* state.

3. The request reaches its due date. In that case, the current message is dropped. The process becomes *Idle* if there are no messages in the buffer, or enters the *Request* state (for a new access request) if there remain pending messages in the buffer.

Having entered the *Scheduled* state, the background process will transmit the current message. For as long as the buffer is nonempty, the process will try to sustain the session (avoiding the contention phase) by posing new bandwidth requests via the signaling channel. If the buffer drains out, the mobile will terminate the session and the background process will enter the *Waiting* state.

In our model, an SMS message is treated exactly as a file transfer, using exactly the same scheduling approach, except that the transfer is scheduled with the lowest possible priority, after all other traffic classes have been accounted for.

The numerical parameters of the SMS model used in our experiments are listed in Table 3.4. Both the interarrival time and message length are exponentially distributed.

3.5 Performance

The performance of our virtual implementation of BRICS has been investigated by simulation and compared to the performance of three other CDMA-based protocols: WISPER [40], VSG-CDMA [37], and S-CDMA (described below). All protocols were implemented in the same virtual radio environment in which the sole criterion of a successful reception was the bit energy to noise density ratio E_b/N_0 (affected by a steady level of background noise) perceived at the receiver. We claim that this kind of environment is fair for a comparison of these particular protocols because they all operate within essentially the same set of prerequisites: power control (and possibly other feedback from the base station) aimed at keeping E_b/N_0 in line with their assumed performance criteria. Consequently, more subtle properties of the radio channel, e.g., errors incurred by non-uniform noise and fading, would affect all protocols to the same degree. The original performance models of WISPER and VSG-CDMA [37, 40] were similar or simpler than our models.

In the figures of the following sections, *Class 1* indicates the voice traffic, *Class 2* indicates the video traffic, *Class 3* indicates the file transfer traffic, and *Class 4* indicates the SMS traffic.

3.5.1 The other protocols

Our virtual implementation of WISPER assumes exactly the same channel parameters as in BRICS, including the basic rate (500 kbps) and the numerical attributes of the frame (the gap, the preamble, the request slot and the regular slot). Consequently, the resulting total frame length consists of 20 384-byte slots, and its back to back time duration is 17.24 ms. The reasons that this protocol is selected for comparison are that, it uses the similar frame structure as BRICS, it uses multiple codes and fixed spreading gain CDMA techniques as in BRICS, this protocol is designed to support multiple services, and it is optimized for throughput.

VSG-CDMA is a variable rate protocol, and its framing parameters cannot be directly related to those of BRICS (because the entire frame in VSG-CDMA is essentially a single slot). The frame (slot) length is equal to the duration of the probability distribution update cycle, which in our model was set to 50 ms. Protocol independent elements (e.g., the power range available to the mobiles, fading properties of the channel, background noise, E_b/N_0 requirements of the receiver at the base) were identical in all models. This protocol is chosen for comparison because it uses variable spreading gain CDMA technique which is different

from fixed spreading gain technique in BRICS.

S-CDMA is a simple protocol without admission control, which we included in our study to illustrate and emphasize the importance of admission control in CDMA-based schemes. Time is divided into equal length slots, with each slot carrying 10 packets (ATM payloads). Each mobile has its own pre-assigned CDMA code. A packet transmitted by a mobile is lost if its received E_b/N_0 ratio is below the acceptable level. The base station provides a simple power feedback to the mobiles. When the base senses that the received E_b/N_0 ratio is over a certain threshold P_h, it commands the mobile to reduce its transmission power by a prescribed factor δ_s. Similarly, if the perceived E_b/N_0 is below another threshold P_l (but the packet can still be received), the base will request the mobile to raise its power by the same factor δ_s. For as long as the bit energy to noise density remains between P_l and P_h, the mobile is allowed to retain its current power. After a silent period, during which there has been no feedback from the base, the mobile uses its last used transmission power level as the initial value. After a failed transmission attempt, the mobile increases its power by δ_s and tries again.

3.5.2 Bandwidth utilization

In this following experiments, the number of mobile stations is always the same for all four traffic classes, with the load level (the horizontal axis) indicated by the number of mobiles (within each class). For example, the load of 80 means that there are 80 voice mobiles, 80 video mobiles, and so on.

3.5.2.1 BRICS

Figure 3.9 illustrates how the link bandwidth in BRICS is shared among the four traffic classes described in Sections 3.4.4–3.4.7. This performance graph has been obtained for the total spread bandwidth $W = 20\,\text{MHz}$.

With the voice traffic having the highest priority, the total amount of bandwidth occupied by voice sessions increases exactly linearly with the increasing number of voice mobiles (the system capacity is much higher than 150 voice sessions). The part of bandwidth assigned to video sessions (which have the second highest priority and the largest bandwidth requirements) increases linearly until about 25 mobiles, when the system becomes saturated with voice and video traffic. From then on, the video sessions yield bandwidth to voice sessions.

Note that although file transfers and SMS sessions have lower priorities than video ses-

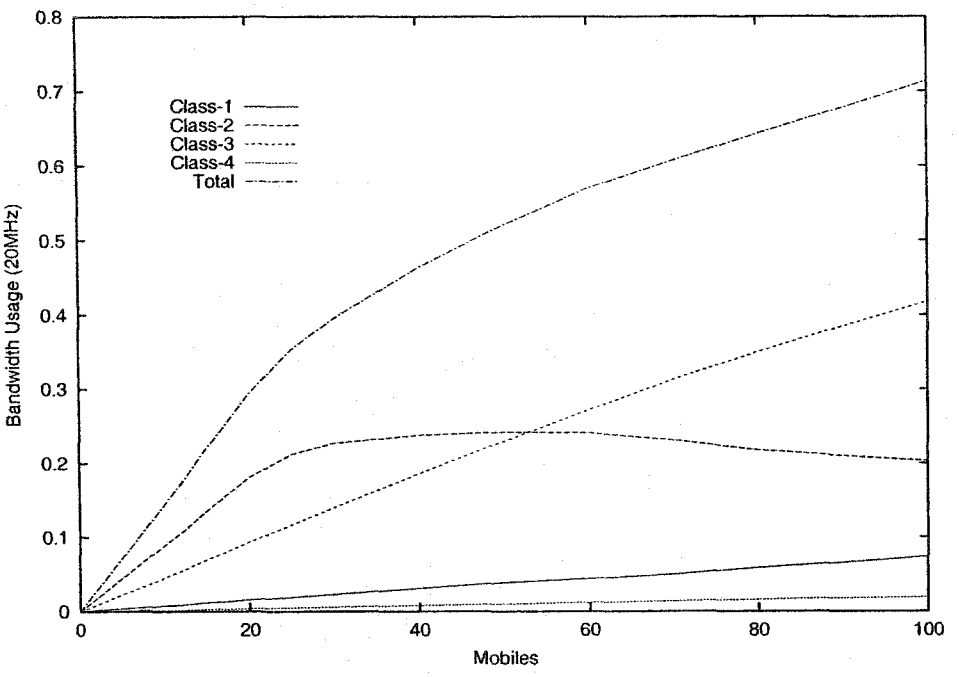


Figure 3.9: Bandwidth utilization in BRICS ($W = 20\,\text{MHz}$)

sions, their assigned bandwidth tends to increase until the very end of the investigated range of traffic conditions. There are two reasons for this behavior. First, there exist silent periods in voice sessions that cannot be reused to set up new voice or video connections, but are freely available to file transfer and SMS traffic classes. Moreover, because of the inherent variability in video load (combined with intermediate buffering—Section 3.4.5), an equivalent of silent periods also does occur in video sessions. Second, the bandwidth requirements of a video session are stringent and must be granted in multiples of TF channels. Consequently, there are bound to exist chunks of bandwidth unusable by video sessions, but available to the much less picky file and SMS transfers.

The QoS tradeoffs in BRICS are somewhat better visible in a network with smaller total spread bandwidth W, which leaves less room for the low priority traffic to sneak in. Figure 3.10 has been obtained for a network with $W = 5\,\text{MHz}$. One can clearly see in it how all three lower priority classes yield to voice.

WISPER

The performance of WISPER under identical offered load is shown in Figure 3.11. The protocol is optimized for throughput, and this results in a bandwidth distribution pattern that is directly related to the traffic pattern. That is, Class-2 traffic (video) consumes the largest amount of bandwidth. Following Class-2 is Class-3 traffic (file transfers), which has

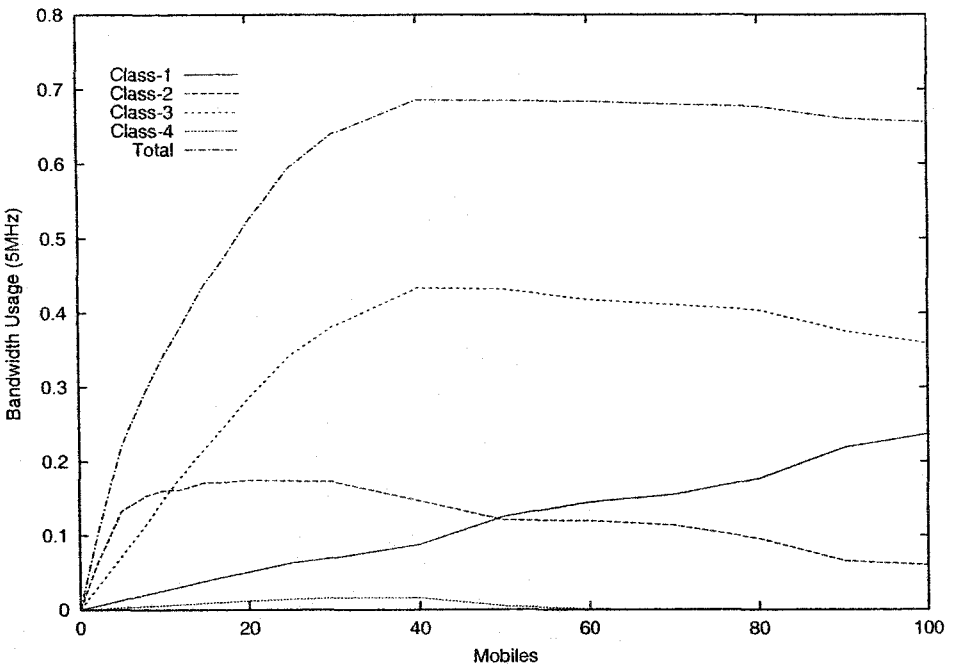


Figure 3.10: Bandwidth utilization in BRICS ($W = 5\,\text{MHz}$)

the second largest offered volume. Class-1 traffic (voice) takes the third place. Finally, Class-4 (SMS) contributes the least amount to the overall load, and it takes the smallest portion of bandwidth.

Notably, the protocol achieves lower maximum channel utilization than BRICS, and the total bandwidth utilization tends to drop when the system becomes saturated. This major trend in throughput is followed by all traffic classes (except SMS, whose bandwidth is too small to be significant), which means that the prioritization of the four traffic types does not fulfill its purpose very well. For example, an increase in the number of admitted video sessions results in a drop in the portion of bandwidth effectively available to voice sessions. This is caused by two properties of WISPER: the rigidity of bandwidth allocation, which requires that a given time slot be filled with traffic of the same class, and the relatively poor performance of the contention resolution part of the protocol (whose impact was neglected in the original analysis of WISPER presented in [40]). It is for the second reason that the throughput of WISPER drops to zero when the system is heavily overloaded (as shown in Figure 3.12).

Looking at Figure 3.11, we can see the different characteristics of the bandwidth curves corresponding to different traffic classes. For as long as the total demand for bandwidth remains below the system capacity, each curve grows steadily following the increase in the

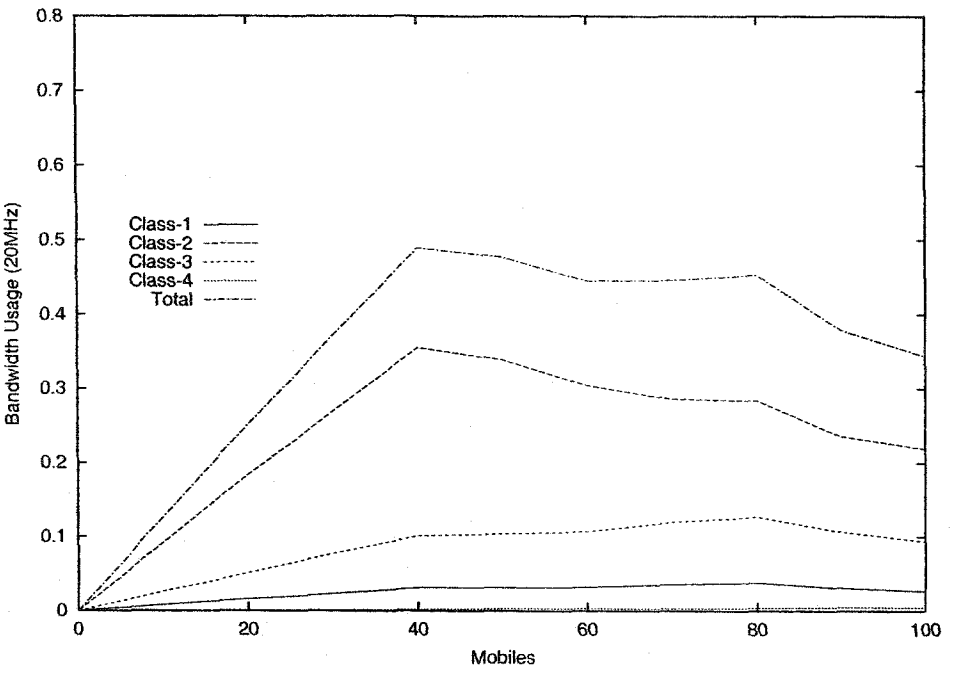


Figure 3.11: Bandwidth utilization in WISPER ($W = 20\,\text{MHz}$)

number of active mobiles. When the system reaches the maximum capacity of the random access channel, the failure rate of access requests increases sharply—as shown in Figure 3.13. For some time, despite the heavy contention and the large number of access failures, sufficiently many mobiles make it to the base to sustain the growth of bandwidth utilization. However, when the collision rate exceeds 85%, the bandwidth assigned to video sessions begins to drop due to the reduced number of video access requests arriving at the base station. According to the scheduling scheme used by WISPER, video requests receive a high priority because of their combination of heavy bandwidth demands with very short deadlines. In contrast, voice requests, although they also have short deadlines, pose considerably lower bandwidth demands than video. File transfers are similar to voice in this respect: although they can use a lot of bandwidth, their deadlines are very long. In consequence, when the system reaches its capacity, there are more voice and file transfer requests queued at the base (waiting for bandwidth) than video requests. When the incoming video requests are blocked due to heavy contention, the queued voice and file transfer requests get more opportunities to receive service. This trend is visible in in Figure 3.14 as the slowing rate of increase (decreasing first derivative) of the drop rates for voice and file transfers. The slowing-increase parts of the drop curves for voice and file transfers coincide with the range of traffic conditions when the video requests begin to lose bandwidth. Thus, in Figure 3.11,

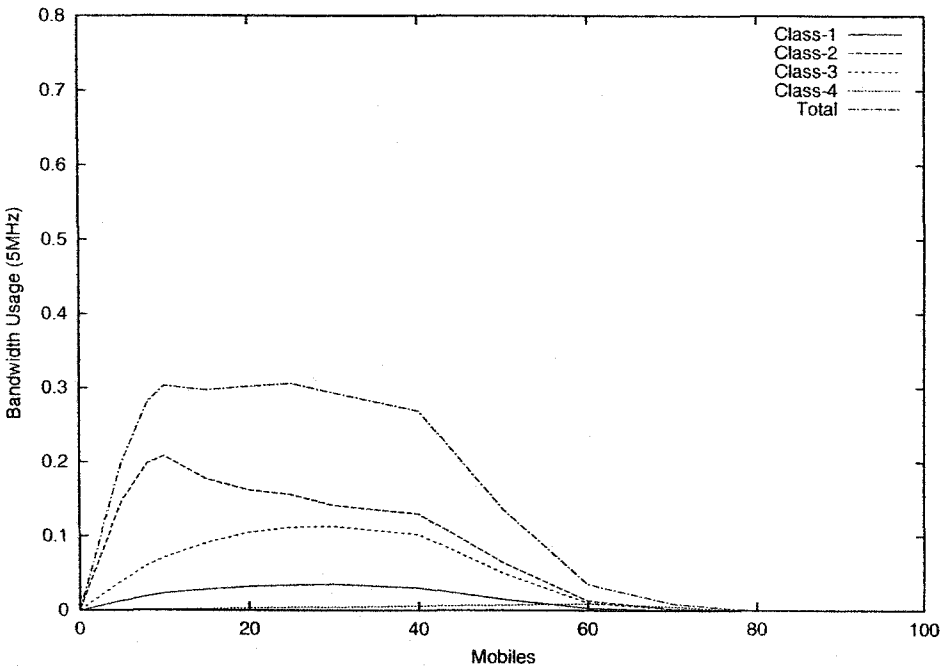


Figure 3.12: Bandwidth utilization in WISPER ($W = 5\,\text{MHz}$)

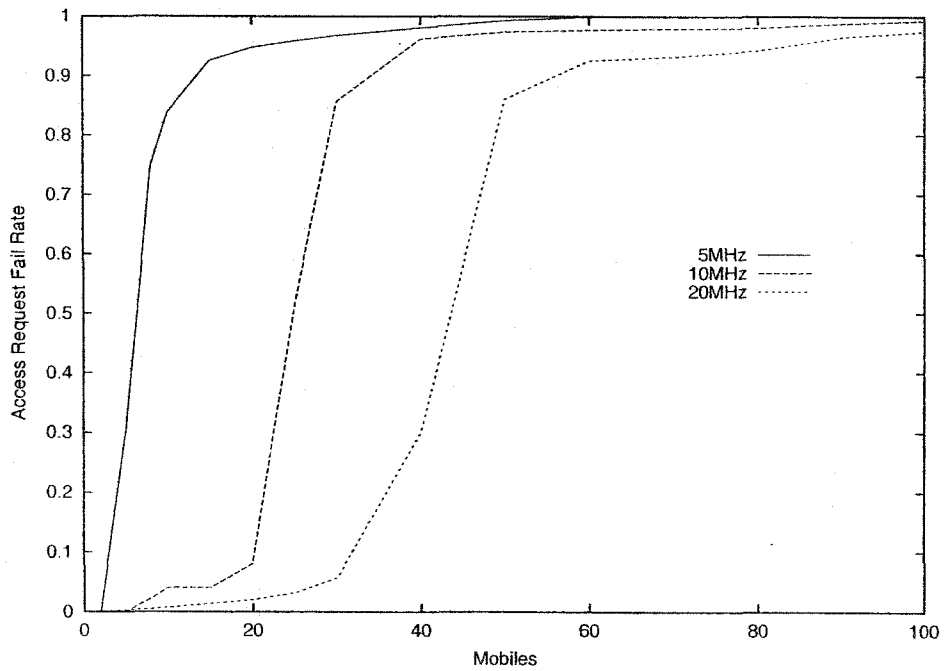
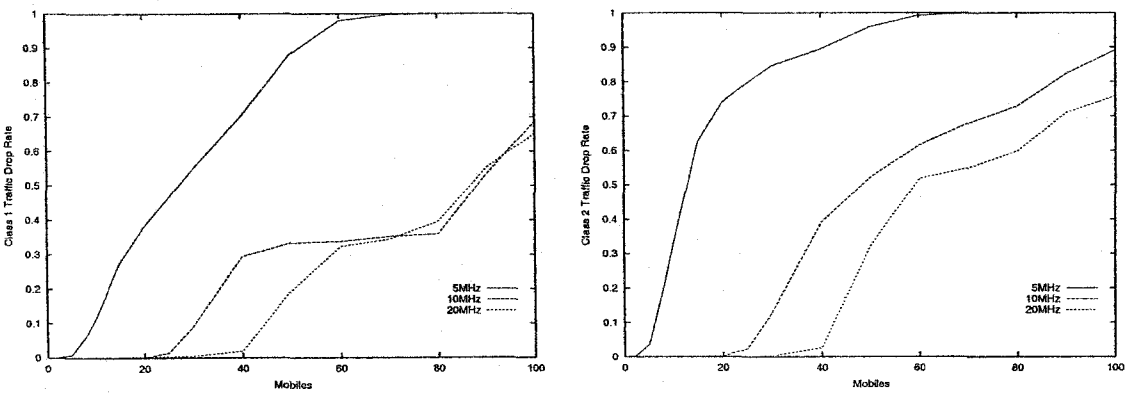
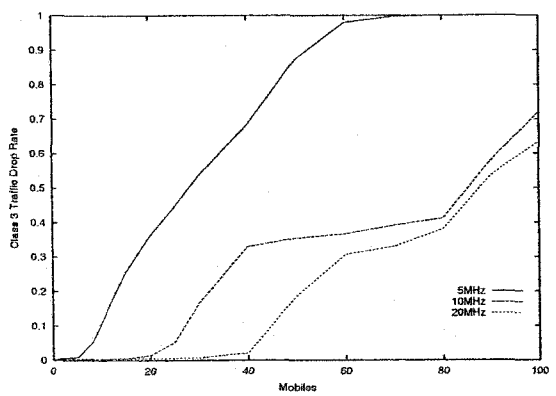


Figure 3.13: Failure rate of access requests in WISPER

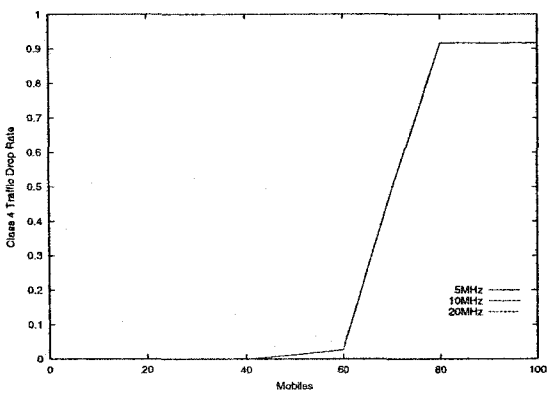


(a) Voice packet drop rate

(b) Video packet drop rate



(c) file transfer packet drop rate



(d) SMS packet drop rate

Figure 3.14: Packet drop rates in WISPER

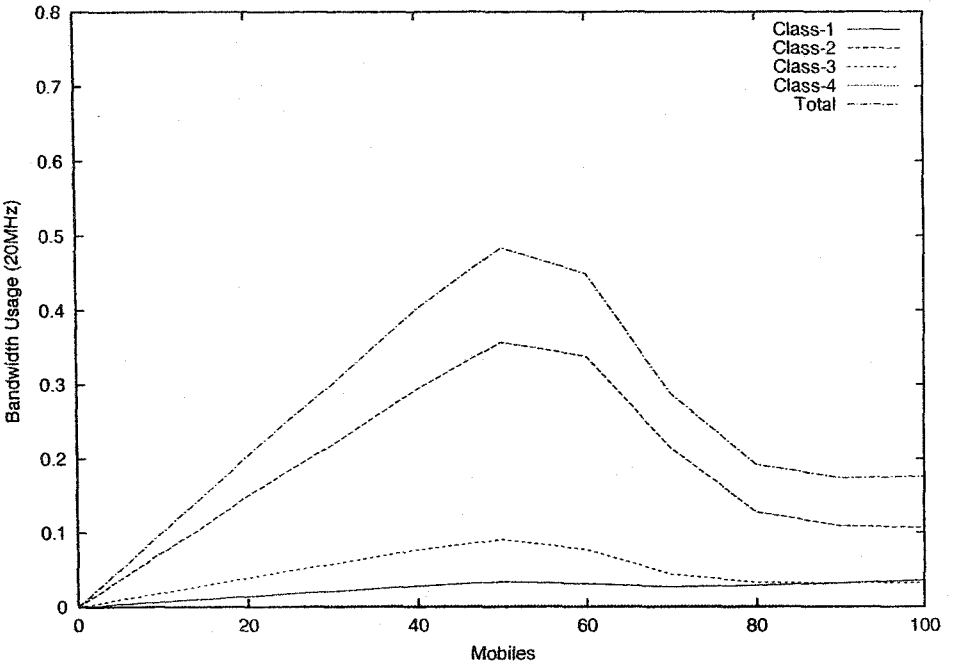


Figure 3.15: Bandwidth utilization in VSG-CDMA ($W = 20\,\text{MHz}$)

the curves for voice and file transfer session stay relatively flat, while the video bandwidth usage starts dropping. With the further reduction in accepted video traffic, the bandwidth used by voice and file transfers even goes up for a while, until the collision rate becomes so high that everybody must pay the price. At this point, the number of successful access requests from all the three traffic classes is reduced to a very low level and their bandwidth usage drops indiscriminately.

It is hard to see the bandwidth usage pattern for SMS class, especially in networks with high spread bandwidth ($W \geq 10\text{MHz}$), because of the small bandwidth demands of this traffic class. In Figure 3.12 ($W = 5\,\text{MHz}$), we can see that the SMS bandwidth usage increases while the throughput of all other traffic classes drops. This is because SMS sessions have long due dates. Once an SMS access request arrives at the base station and it cannot be serviced immediately, it will tend to remain queued for a very long time. The higher the number of SMS mobile stations, the more SMS access requests will accumulate at the base station under heavy overall load. These SMS requests will be serviced when the number of access requests from high priority mobile stations is reduced due to heavy contention and the high rate of collisions of their access requests.

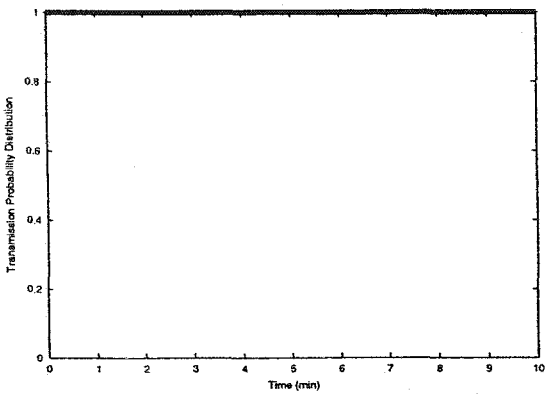
3.5.2.2 VSG-CDMA

A similar tendency for the bandwidth utilization to drop under heavy load can be observed in VSG-CDMA. In fact, this trend is even more pronounced in this case, as shown in Figure 3.15. At first sight this is surprising because the method of adjusting the transmission probability in VSG-CDMA is intended to admit the right amount of traffic to the system, even under heavy contention.

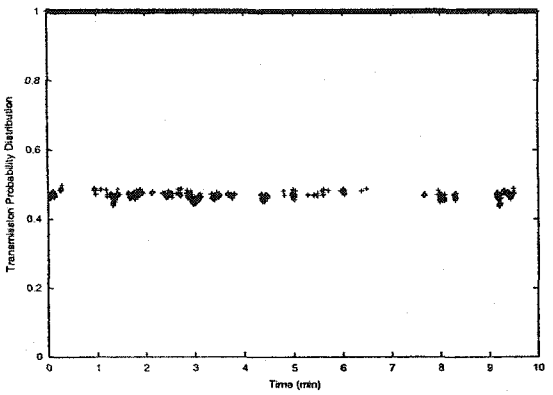
However, as it turns out, a realistic implementation of VSG-CDMA exhibits instabilities. Under light load, the network operates without transmission control, and all ready stations transmit with probability 1. When the load becomes heavy, the system has the tendency to switch between two states. Having detected a congestion in one frame, the base station turns on the controlled mode. Consequently, in the next frame all active users transmit with some probability p, which solely depends on the number of active mobiles. That frame tends to be underloaded, which forces the network back to the uncontrolled mode, which in turn causes congestion, and so on. These oscillations do not necessarily occur on a frame-by-frame basis, but they are clearly visible and so is their impact on the overall performance of the scheme. With the increasing number of active mobile stations, the bandwidth usage initially increases; then, after reaching the maximum, its drops and stabilizes at some point.

Figure 3.16 shows how the behavior of VSG-CDMA is determined by the transmission probability. When the system load is set to 30 mobiles, the total offered load is below the system capacity, and the transmission probabilities are always 1. Within this range of traffic conditions, the bandwidth usage for each traffic class increases in a straightforward manner along with the increasing load (the number of active mobiles). When the offered load reaches 50 mobiles, the network enters the state when it occasionally becomes overloaded and the probabilistic congestion control mechanism becomes effective. The transmission probabilities are still mostly 1, but they sometimes drop to values around 0.48, which throttles down the overall throughput. Additionally, owing to the high level of interference caused by the increased number of active mobile stations (which lowers the ratio of E_b/N_0), the reception failure rate begins to raise sharply (Figure 3.17). All these factors decrease the amount of bandwidth used by all traffic classes.

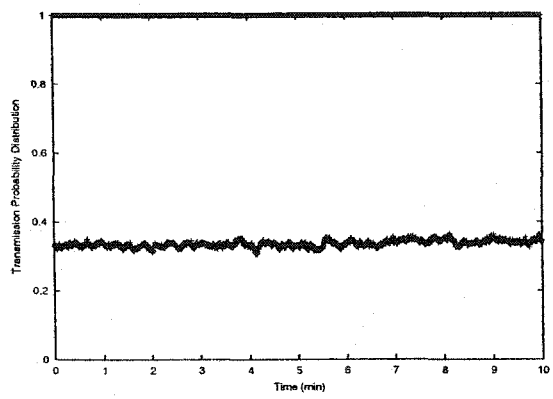
When the system load reaches about 70 mobiles, the low end of the observed range of transmission probabilities moves down to values centered around 0.33, and those probabilities appear to be more focused around this value. This causes the bandwidth usage to decrease even further. When the offered load approaches 90 mobiles, the low end of the



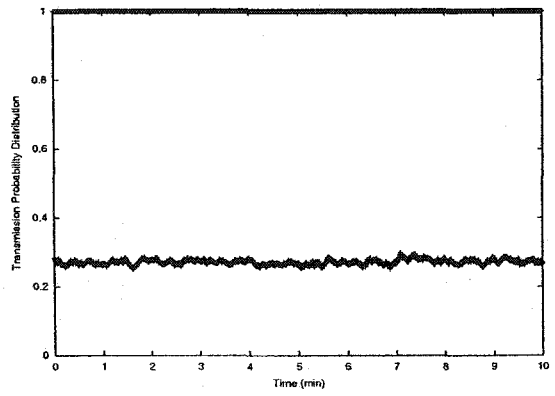
(a) Distribution at system load 30



(b) Distribution at system load 50



(c) Distribution at system load 70



(d) Distribution at system load 90

Figure 3.16: Transmission probability distribution in VSG-CDMA ($W = 20\,\text{MHz}$)

transmission probabilities decreases to around 0.26. Although the transmission probability is now quite low, the bandwidth usage of video and file transfer sessions remains relatively constant, due to the high number of active mobiles in the system.

The amount of bandwidth utilized by voice sessions begins to drop after the system load crosses the 70 mobile mark. This is because, in this system, voice traffic has the lowest transmission rate under the same spread bandwidth and the highest spreading gain. Also, voice service has the lowest E_b/N_0 requirements, which can be easily met at the base, especially with the reduction in the total number of packet transmissions. As the number of active mobile stations increases, the two counteracting factors, i.e., the increase in the offered load and the decrease in transmission probability, stabilize each traffic class at its specific constant level of bandwidth utilization which, however, is considerably lower than the maximum.

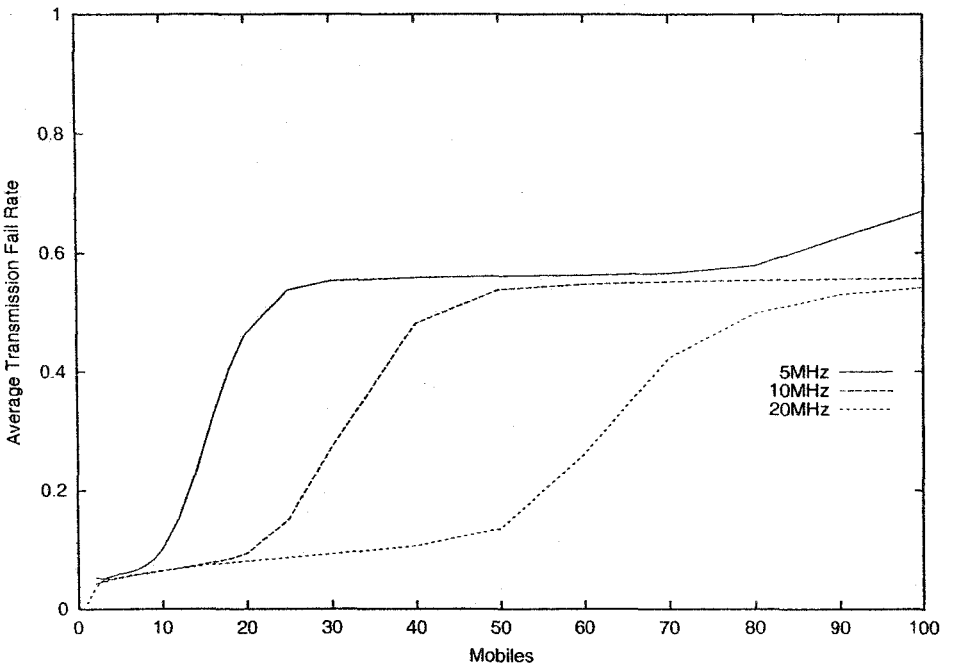


Figure 3.17: Transmission failure rate in VSG-CDMA

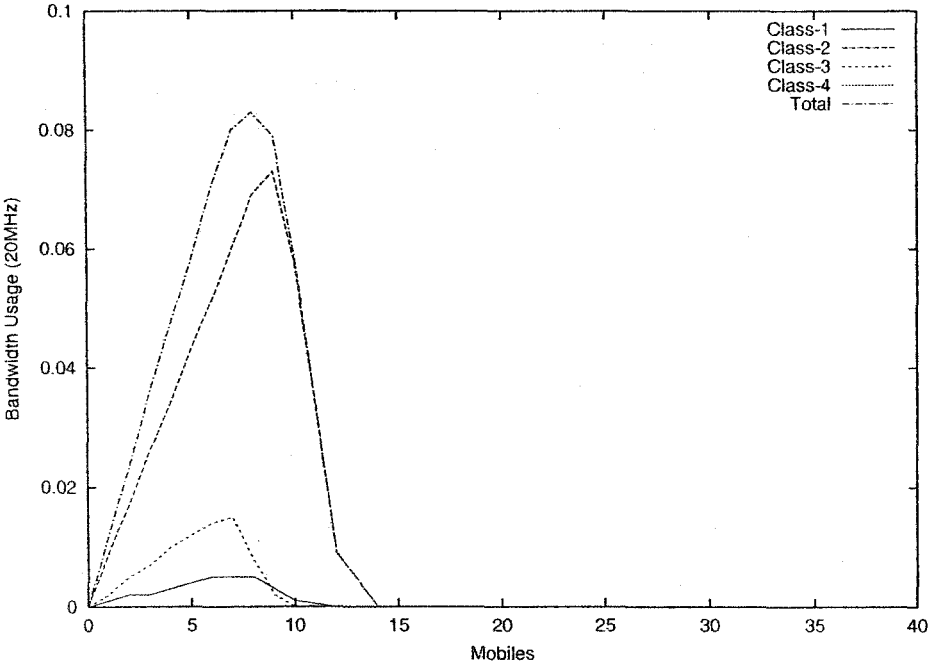


Figure 3.18: Bandwidth utilization in S-CDMA ($W = 20\,\text{MHz}$)

3.5.2.3 S-CDMA

The importance of admission control is well illustrated in Figure 3.18 obtained for S-CDMA. Although the protocol achieves a relatively high maximum bandwidth at about 7 mobiles in each traffic class, its performance breaks down completely and abruptly when the offered load exceeds the saturation point of the network, considerably more so than in VSG-CDMA. This happens when the system load is around 10 mobiles, and the network practically stops to deliver any traffic at all when the load is over 14. In term of bandwidth utilization, Class-2 (video) takes the first position, Class-3 (file transfer) takes the second place, Class-1 (voice) is third, and Class-4 (SMS) consumes the least bandwidth. Video traffic grabs the largest share of bandwidth because it has the largest offered traffic volume and the lowest E_b/N_0 requirement.

3.5.3 Quality of service

The objective of this series of simulation experiments is to determine how the QoS received by each traffic type is affected by its co-existence with other traffic types. To do this, we set the number of mobile stations within a selected class (or selected classes) to be a constant high value while the load contributed by the remaining classes is increased steadily starting from zero.

3.5.3.1 Fixed load from voice and video

To see how the QoS received by high priority sessions in BRICS is affected by the presence or absence of other traffic types, we carried out a series of experiments in which the offered voice and video load remained steady, while the contribution of the remaining two traffic classes varied. The combined load of voice and video was set at a high level—to make the drop rate (and any deviations thereof) clearly visible, while the file transfer and SMS load increased in proportion to the number of mobiles in the network. Figure 3.19 illustrates the stability of the high-priority service in BRICS.

While the amount of bandwidth used by voice and video sessions is affected to little extent by the low priority sessions, there is a perceptible drop in the video bandwidth and a slight, almost imperceptible, increase in the voice bandwidth under 20 MHz spread bandwidth. These phenomena have their source in the varying level of contention on the *RA* channel, as the intensity of file transfers and SMS traffic becomes higher. With some bandwidth being reserved by video sessions, a few voice requests may remain queued for a while at the base station before being blocked—until their short deadlines expire. The

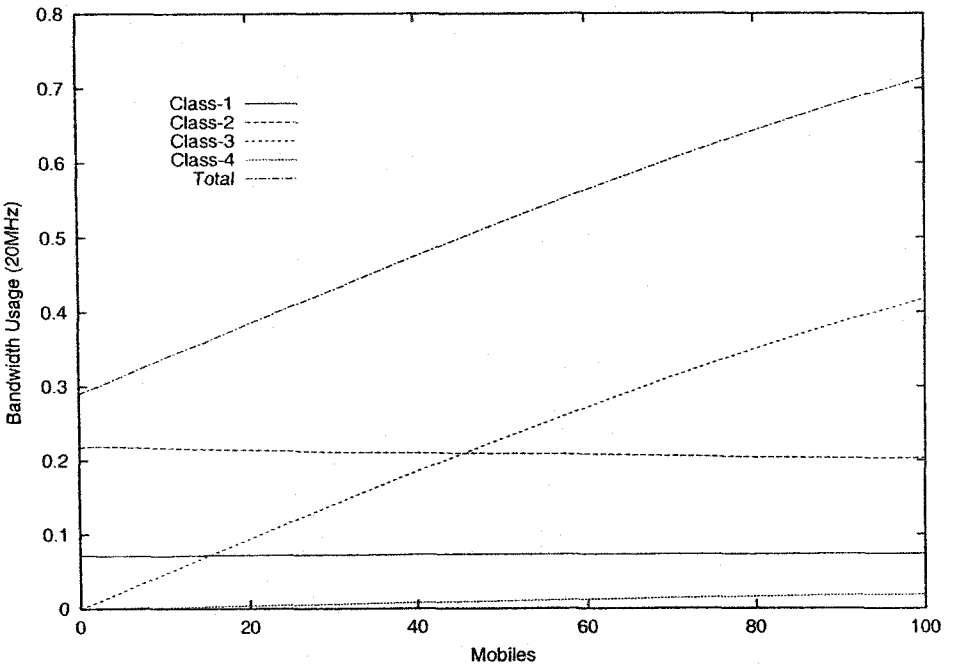
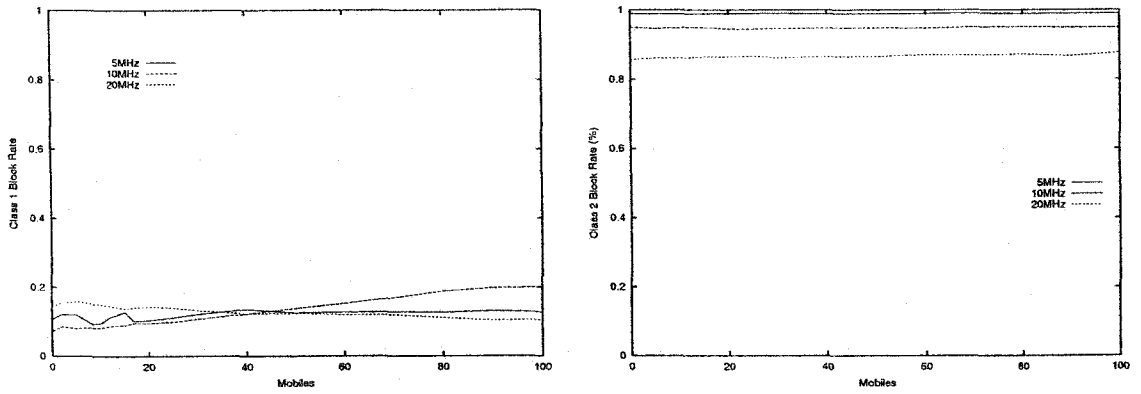


Figure 3.19: Bandwidth utilization in BRICS under fixed voice and video load (W = 20 MHz)

increasing number of file transfer and SMS requests contribute to the contention on the *RA* channel. According to the contention permission scheme, if the interference level in the *RA* slot is over the threshold, lower priority requests are denied access until the higher priority access queues at the base are empty. This will slightly reduce the population of video sessions in the system, as indicated in Figure 3.19 by the perceptible steady drop of the video bandwidth, and in Figure 3.20 by the increasing blocking rate of video under 20MHz. With the slight reduction in the bandwidth reserved for video sessions, more voice sessions queued at the base are accommodated, and the blocking rate for voice service is reduced (as shown in Figure 3.20, the 20MHz case). The impact of this phenomenon can be somewhat influenced by adjusting the deadline of voice sessions and the contention threshold for the *RA* channel. Note, however, that the reduction in the video throughput does not match the increase in voice throughput. This is because different contention opportunities translate into different bandwidth utilization patterns for the two traffic types. In particular, a sizable fraction of the bandwidth reserved for a voice session goes to its silent periods, which part is reused by file and SMS transfers.

By comparing Figures 3.19, 3.21, and 3.22 we can see that under fixed load from both voice and video mobiles, the percentage of bandwidth assigned to voice service increases with the reduction in spread bandwidth (20MHz, 10MHz, and 5MHz), while the bandwidth



(a) Voice block-rate

(b) Video block-rate

Figure 3.20: Voice and video block-rate in BRICS under fixed voice and video load

share taken by video sessions decreases. In the 20MHz case, video sessions take more than 20% of the overall bandwidth, and voice receives less than 10%. In the 10MHz case, the two traffic classes are very close together while for W = 5MHz, voice prevails with more than 20% and video receives less than 10%.

Let us start from the high end, i.e., the 20MHz case. Until the access contention from file transfers and SMS sessions becomes heavy, the load on the RA channel is low, and the voice and video requests can easily make it to the base station. Since a significant amount of bandwidth ends up reserved by video sessions, the voice blocking rate is a little bit higher at the initial range of traffic conditions (see Figure 3.20). Since the voice service only takes a small portion of the total bandwidth (in the 20MHz case), the video bandwidth usage is limited by the bandwidth reservation from both voice service and video service.

In the 10MHz case, due to bandwidth limitations, the number of access requests queued at base station from each traffic class frequently exceeds its contention permission threshold. Also the RA channel tends to be overloaded when the number of mobiles involved into file transfer and SMS sessions becomes high. Consequently, the contention permission thresholding is employed more frequently, which has the effect of reducing the failure rate of access requests—as shown in Figure 3.23. The same mechanism also reduces the number of active video sessions in the system, even when the load from file transfer and SMS mobiles is low. This is why the video service takes a smaller portion of the the total bandwidth in comparison to the 20MHz case. With less bandwidth reserved by video, more voice requests can be serviced, and this is why voice takes a larger fraction of the bandwidth compared to the 20MHz case. This is also the reason why the voice blocking rate for W = 10MHz is

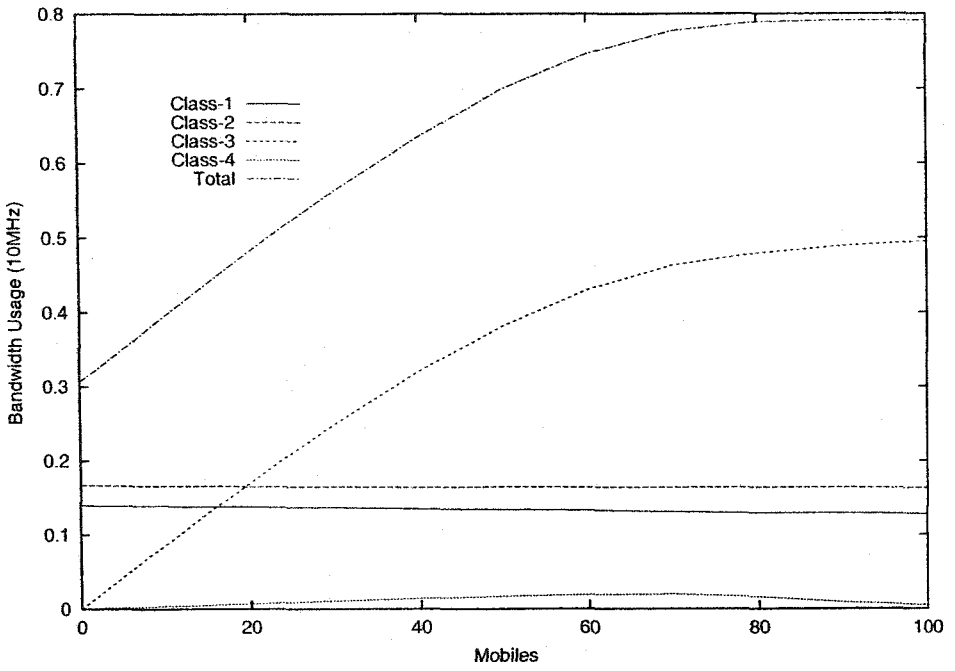


Figure 3.21: Bandwidth utilization in BRICS under fixed voice and video load (W = 10 MHz)

low in the initial range of traffic conditions—in Figure 3.20. Also in this case, as explained for $W = 20$ MHz, the bandwidth assigned to video sessions drops as the contribution from file transfers and SMS sessions becomes more pronounced, but this decrease is too small to increase the share of voice sessions. Thus, the blocking rate for the voice mobiles keeps going up, and their bandwidth usage drops slightly, with the increasing load from file transfers and SMS sessions.

In the 5MHz case, contention permission thresholding is applied even more often than for $W = 10$ MHz. This leads to a situation whereby a considerable number of lower priority mobiles are denied access to the system. This behavior also affects video requests, as shown in Figure 3.20, and is responsible for the unexpectedly low collision rate under heavy load from file transfers and SMS sessions (see Figure 3.23). Voice mobiles also tend to be occasionally denied contention permission when their queue for bandwidth at the base station becomes too long. Additionally, they may be blocked due to high collision rate in the RA channel. This makes the voice blocking rate increase at the beginning (Figure 3.20) and then remain relatively flat, as the contention permission thresholding kicks in to throttle the collision rate down to a low stable level (Figure 3.23).

Because of the drastic limitation on bandwidth (5MHz) and the heavy contention on the RA channel, video requests are primarily limited by contention permission control.

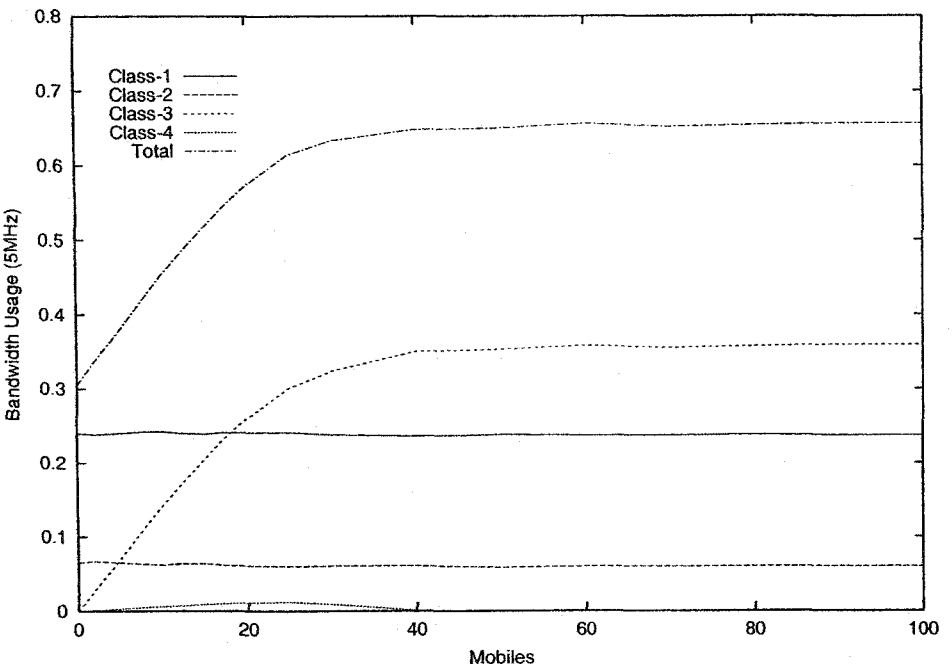


Figure 3.22: Bandwidth utilization in BRICS under fixed voice and video load ($W = 5\,\text{MHz}$)

This prevents the video sessions from reserving a lot of bandwidth. Consequently, a voice request stands a good chance to be serviced as soon as it is submitted. Thus, voice sessions take more bandwidth than video sessions and their blocking rate is even lower than in the 20MHz case, when the number of file and SMS transfers is small. As explained above, more bandwidth becomes available for voice service in the 20MHz case when the number of file transfers and SMS sessions is high.

3.5.3.2 Fixed load from a single class

To see how a single type of service is affected by other services under different protocols, we carried out another series of experiments in which the offered load from one traffic class (voice, video, file transfer, or SMS service) remained steady, while the contribution from the remaining three traffic classes varied. The individual load from the selected service was set at a high level, while the load from all other services increased in proportion to the number of mobiles in the system.

BRICS

We can see quite clearly in Figure 3.24 that other traffic has little impact on voice bandwidth utilization in BRICS. The voice throughput maintains a relatively stable level under each of the three investigated values of the spread bandwidth W. The factor responsible for the

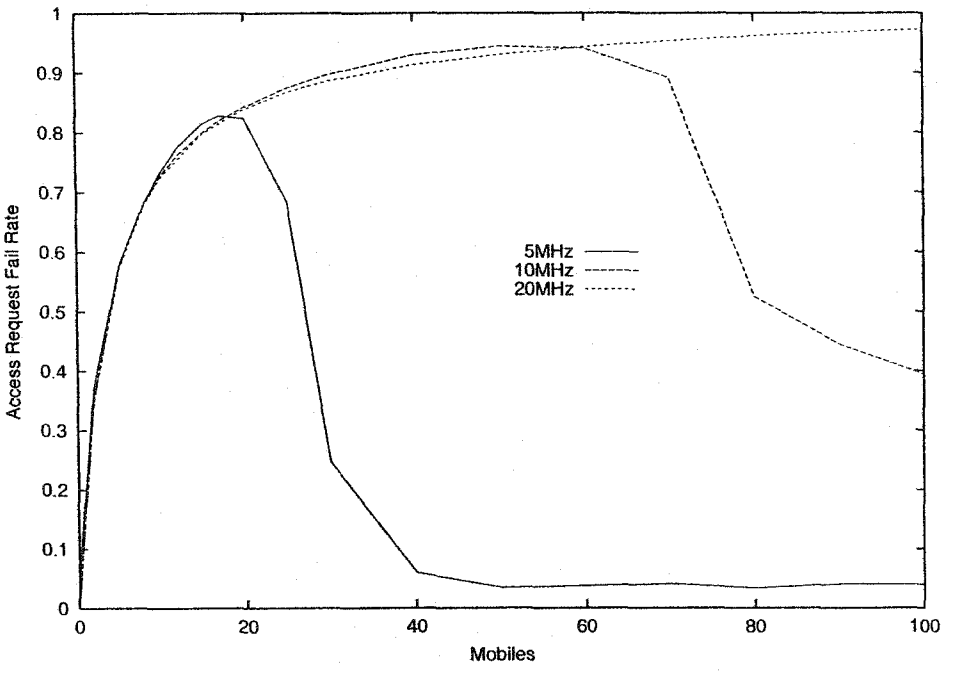


Figure 3.23: Access request failure rate in BRICS under fixed voice and video load

slight drop (almost imperceptible in the case $W = 20\,\text{MHz}$) is the increased contention on the RA channel under heavy load.

Figure 3.25 shows how the bandwidth available to video sessions is affected by other traffic classes in BRICS. The video service has the second highest priority, and its bandwidth utilization is mainly affected by the voice service. The video throughput is reduced quickly with the increase in the number of voice mobiles, which behavior is most pronounced when the total bandwidth is limited, like in the 5MHz case. The increased number of collisions in the RA channel also reduces the throughput of video sessions.

Figure 3.26 shows that the bandwidth usage of file transfer sessions decreases with the increasing load from all other traffic classes, under all three values of the spread bandwidth W. But, the decrease is not as drastic as one could expect, considering that the file transfer service has the third priority. This is because, when the load from voice and video services is high, the file transfers mostly reuse the temporarily released bandwidth (the silent periods) reserved by active voice and video sessions. The level of contention (collisions) on the RA channel also affect the throughput of the file transfer sessions.

The impact of the other traffic classes on SMS is shown in Figure 3.27. Notably, this impact is drastically different, depending on the spread bandwidth W. When the spread bandwidth is limited ($W = 5\,\text{MHz}$), the bandwidth usage of SMS decreases quickly with the increasing number of mobile stations involved in the other session types. This is because

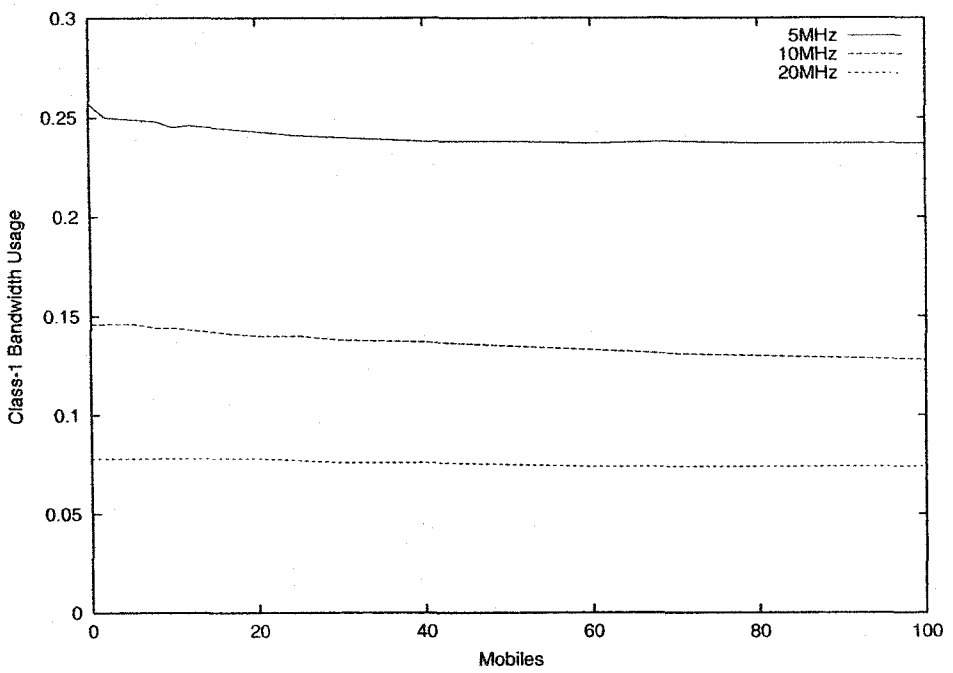


Figure 3.24: Fixed-load voice bandwidth utilization load in BRICS

SMS traffic has the lowest priority. In the 10MHz case, the SMS sessions are affected to a significantly smaller degree, and practically not all for W = 20,MHz. This is because SMS sessions require rather little bandwidth and they can easily squeeze their packets into whatever leftovers remain available after the three higher priority classes have been serviced.

WISPER

As illustrated in Figure 3.28, voice bandwidth in WISPER tends to decrease with the increasing load from the other traffic classes, which behavior is quite drastic when the total spread bandwidth is low (W = 5 MHz). When the spread bandwidth is high (W = 20 MHz), the voice sessions are able to get more of their share, due to their short deadlines, which shows as a relatively slow drop in Figure 3.28.

Although video service has the highest scheduling priority, the poor performance of the contention resolution scheme in WISPER reduces its accessibility to the system bandwidth. The video bandwidth usage keeps decreasing with the increasing load from the other traffic classes—as shown in Figure 3.29.

The file transfer sessions can get their bandwidth due to their heavy offered load. This is shown in Figure 3.30 as the flat portion at the beginning of each curve. As explained in section 3.5.2, the bandwidth usage pattern of file transfers has a similar characteristics to that of the voice service.

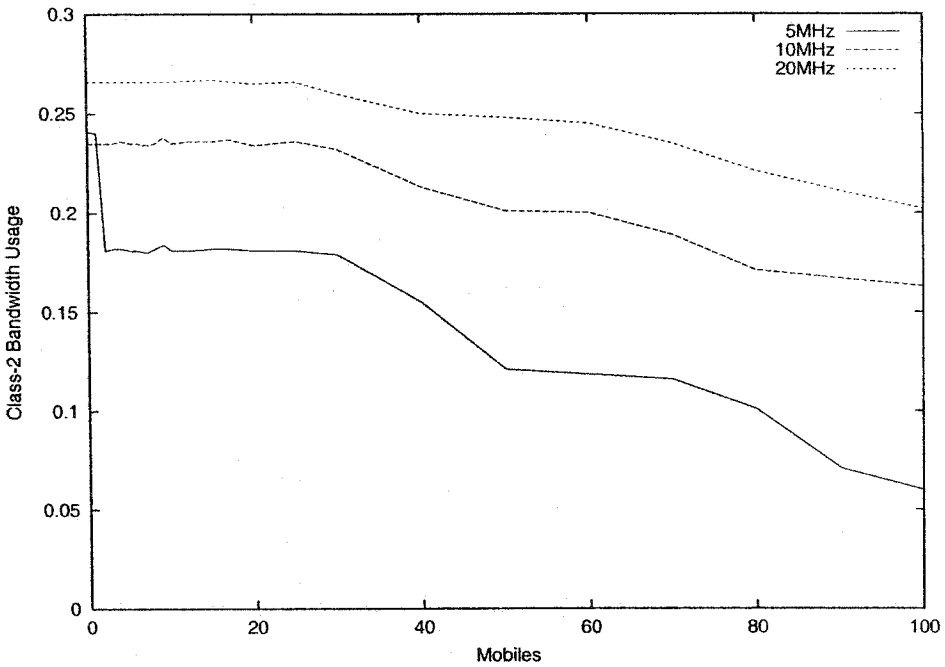


Figure 3.25: Fixed-load video bandwidth utilization in BRICS

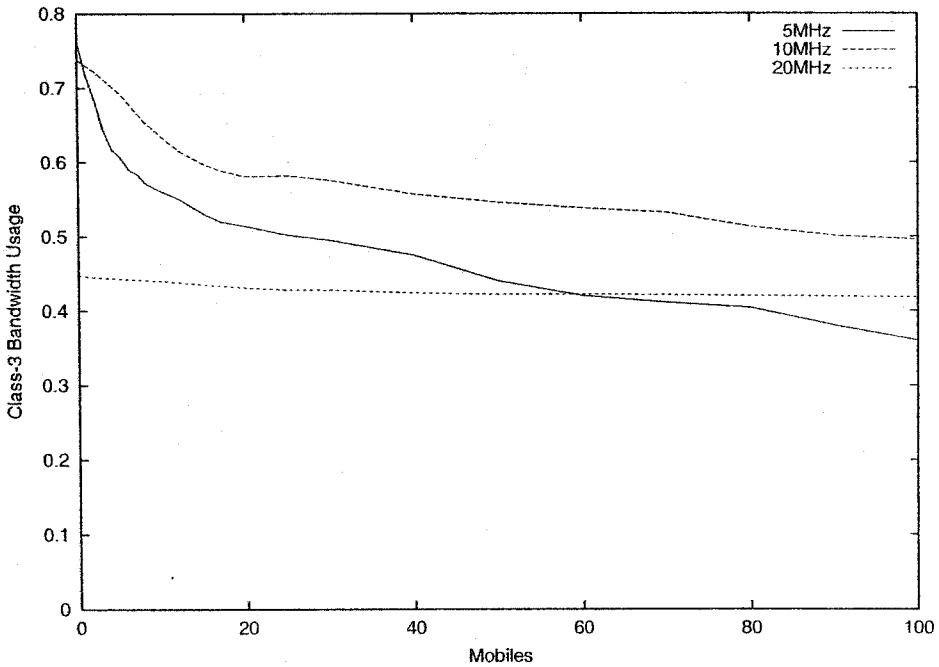


Figure 3.26: Fixed-load file transfer bandwidth utilization in BRICS

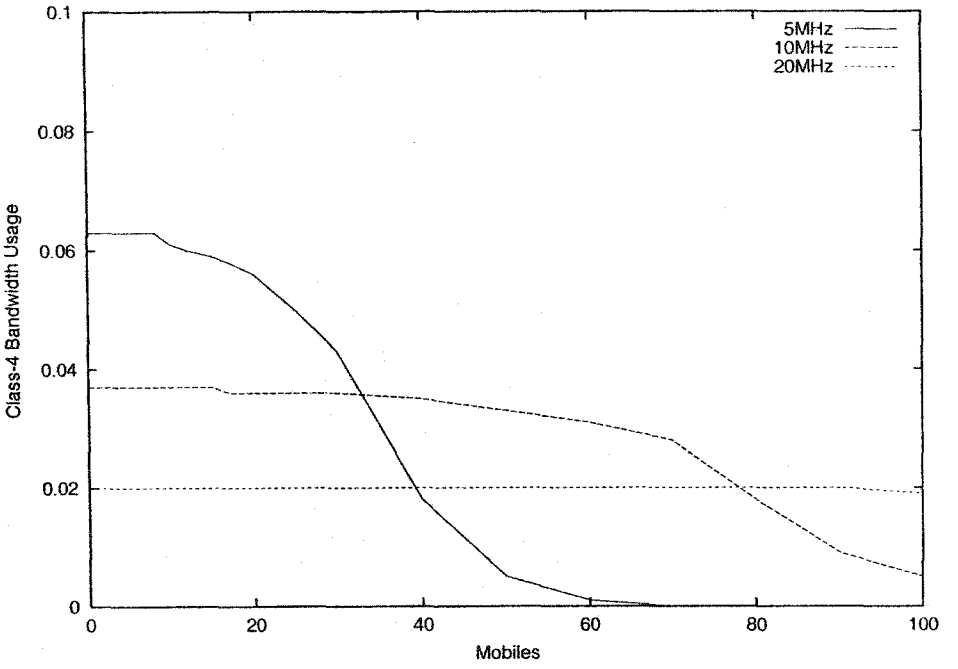


Figure 3.27: Fixed-load SMS bandwidth utilization in BRICS

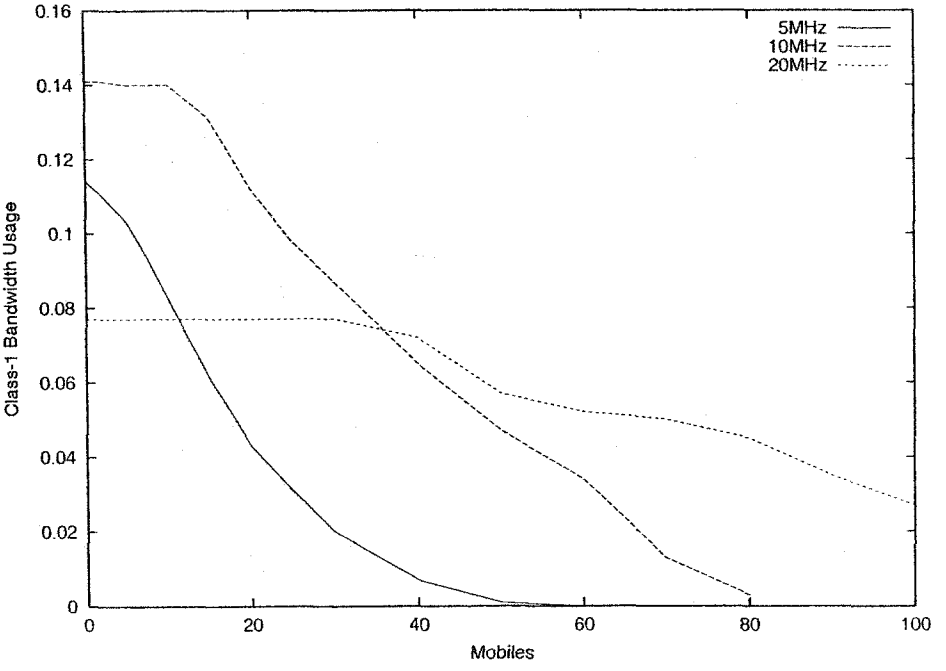


Figure 3.28: Fixed-load voice bandwidth utilization in WISPER

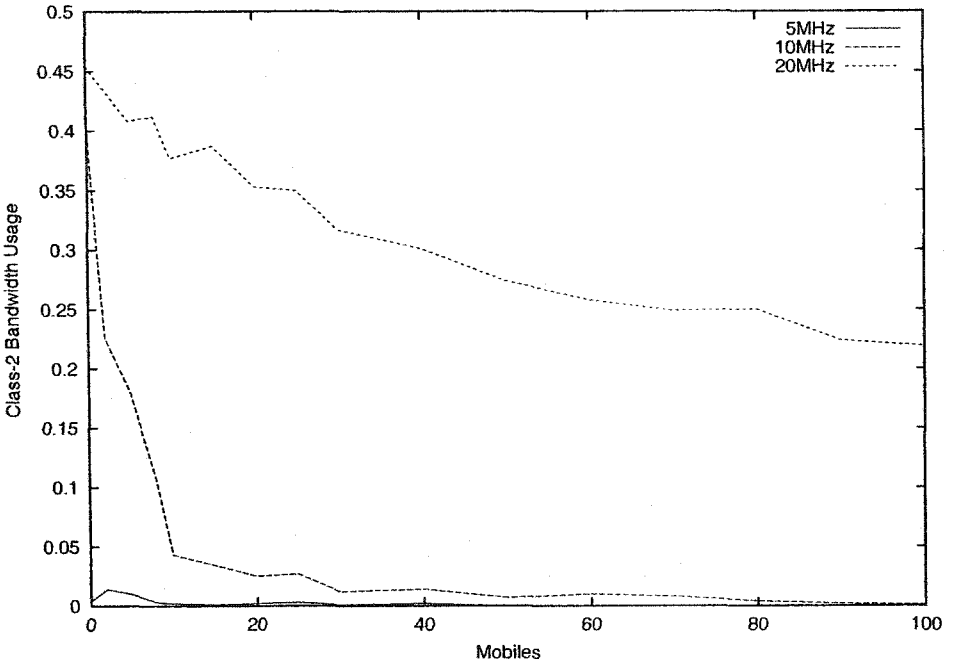


Figure 3.29: Fixed-load video bandwidth utilization in WISPER

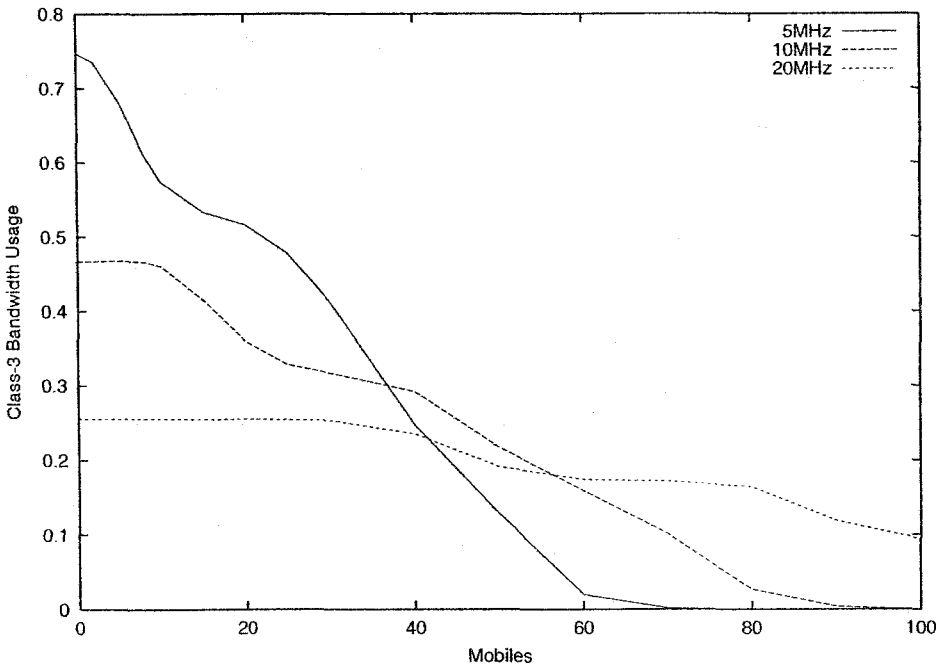


Figure 3.30: Fixed-load file transfer bandwidth utilization in WISPER

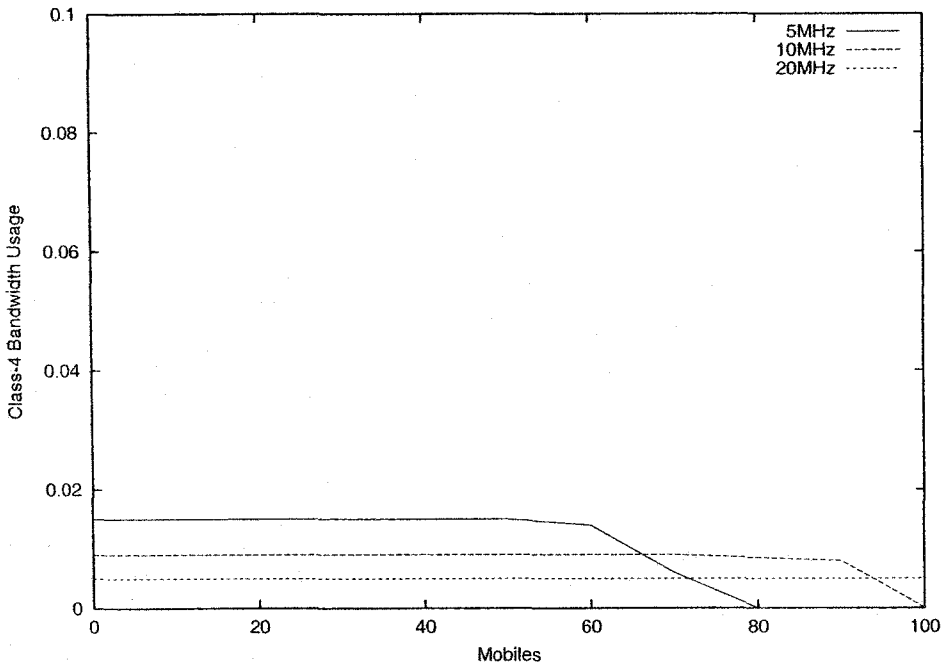


Figure 3.31: Fixed-load SMS bandwidth utilization in WISPER

Due to the long deadlines of SMS packets, load variations affecting other traffic classes have little impact on SMS bandwidth usage. The SMS bandwidth usage only drops when the contention to the *RA* channel is extremely high, as shown in Figure 3.31.

VSG-CDMA

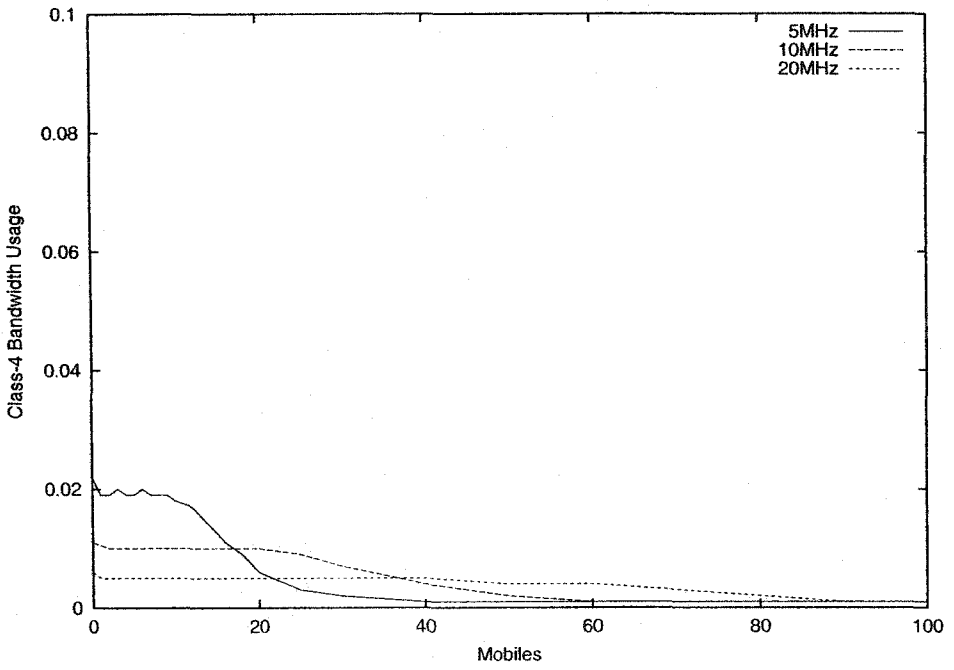


Figure 3.32: Fixed-load SMS bandwidth utilization in VSG-CDMA

Because of its low bandwidth combined with high QoS requirements (in terms of the E_b/N_0), SMS bandwidth usage in VSG-CDMA decreases with the increasing load from the other traffic classes, as illustrated in Figure 3.32. Until light contention, the throughput of SMS sessions remains relatively constant. When the system is overloaded, SMS bandwidth usage drops to a very low level, despite the efforts of the congestion control scheme.

The bandwidth usage of file transfers in VSG-CDMA, shown in Figure 3.33, decreases with the increasing load from the other traffic classes, but it decreases in different proportion depending on the total amount of the spread bandwidth W. The higher the spread bandwidth, the more gradual the drop in the file transfer bandwidth. This can be explained by examining the relationship between the total available bandwidth and the congestion control probabilities.

Figure 3.34 shows three samples of congestion control probability distribution under a fixed load from file transfer sessions. The samples were taken within the drop range of the file transfer throughput for three values of W, i.e., i.e. 5MHz, 10MHz, and 20MHz. Note

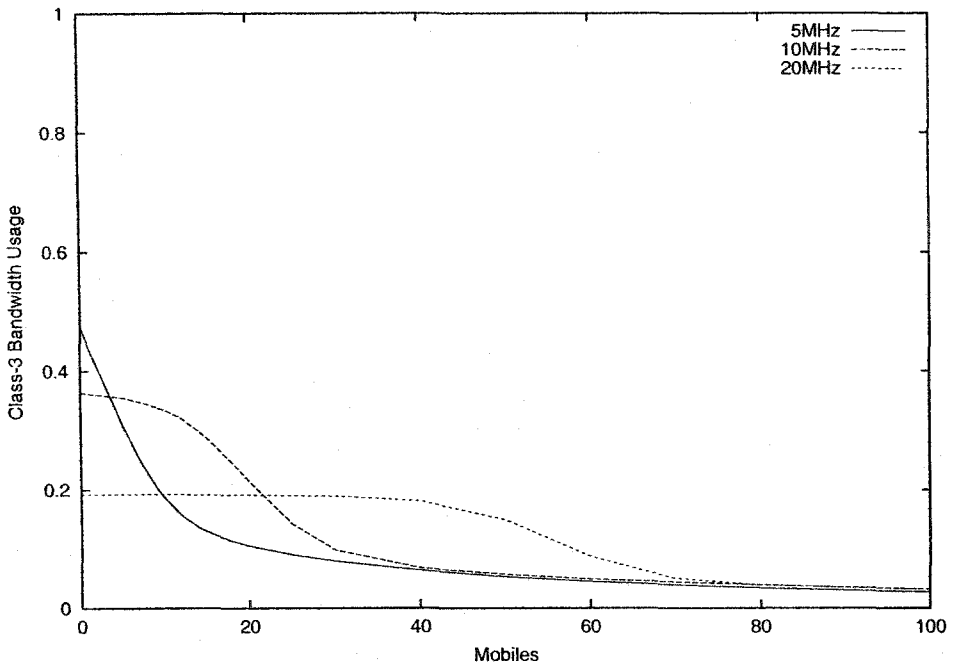
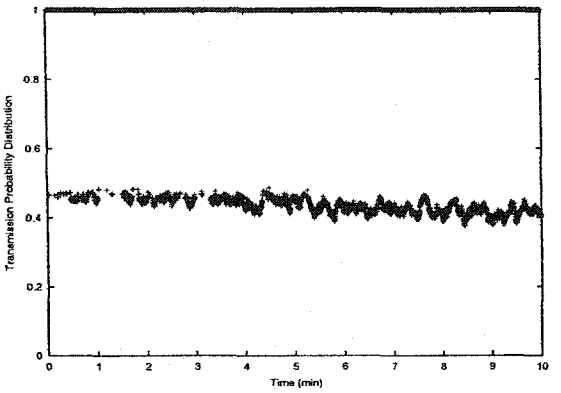


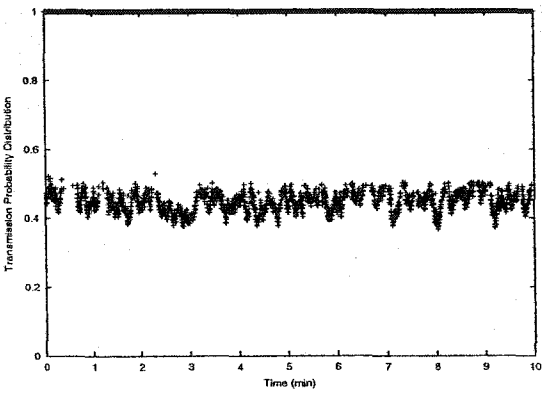
Figure 3.33: Fixed-load file transfer bandwidth utilization in VSG-CDMA

that the congestion control probability distribution spans a wider range for $W = 5\,\text{MHz}$ (Figure 3.34(c)) than for $W = 20\,\text{MHz}$ (Figure 3.34(a)). However, because of the bandwidth limitation in the $W = 5\,\text{MHz}$ case, the transmission failure rate (due to the multiple access interference) increases quickly (see Figure 3.35). This causes the bandwidth usage to drop faster when W is smaller and vice-versa.

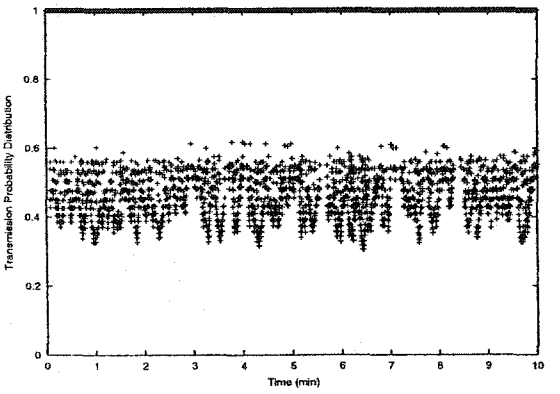
We can see from Figure 3.36 that the bandwidth usage for video traffic shows quite different responses for different W, when the contribution from the other traffic classes is relatively low (less than 30 mobiles). Again, these differences stem from the transmission control probabilities, which are mainly determined by the total system capacity and the traffic load at the moment. For illustration, let us have a closer look at $W = 5\,\text{MHz}$. Figure 3.37(a) shows a ten-minute long (simulated time) sample of the transmission probability distribution obtained under zero load from the other (non-video) types of mobile stations. Under these conditions, the congestion control probabilities lie mostly between 0.19 and 0.8. They concentrate around 0.3 and are sparse in the vicinity of 1. Figure 3.37(b) shows a similar sample obtained under light load from non-video traffic classes (2 mobiles). Now the congestion control probabilities are between 0.19 and 0.7. They concentrate around 0.3 and become even more sparse near to 1. This is why the video throughput drops slightly between 0 and 2 (the 5MHz curve) in Figure 3.36. Figure 3.37(c) shows yet another sample in which the number of non-video mobiles is 25. With the increase in the number of



(a) $W = 20$ MHz, Other traffic load 50



(b) $W = 10$ MHz, Other traffic load 20



(c) $W = 5$ MHz, Other traffic load 5

Figure 3.34: Transmission probability distribution under fixed file transfer load in VSG-CDMA

other mobiles, the congestion control probabilities shift strongly toward the low end. But the number of samples in the vicinity of 1 increases considerably. This is the main reason why the video bandwidth usage increases between 2 and 25 mobiles (the 5MHz curve) in Figure 3.36. With the further increase in the number of non-video mobiles, the congestion control probabilities move further down, and become more focused around a single value, as shown in Figure 3.37(d) (obtained for 50 non-video mobiles). This kind of probability distribution has the tendency to toggle the system between two states: underloaded and overloaded. In the first state, the transmission probability is set to 1. After a while the system becomes overloaded and, in the next interval, the transmission probability is set to a very low value. Then in turn the system becomes underloaded and the scenario repeats itself. These oscillations are the the main drawback of VSG-CDMA. At high probability

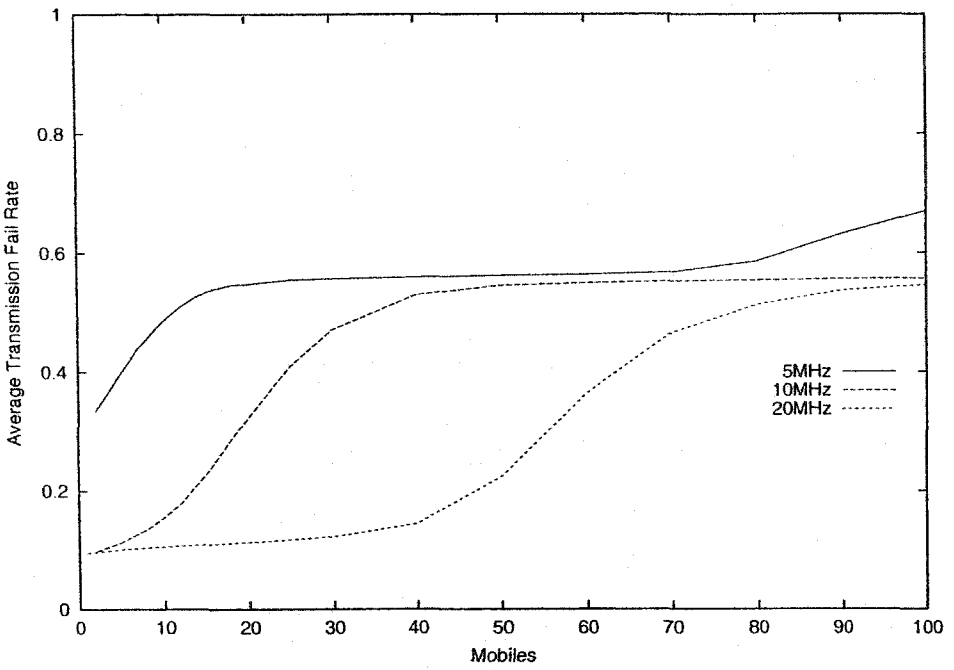


Figure 3.35: Transmission fail rate in VSG-CDMA under fixed file transfer load

of transmission, the video throughput drops because of a high collision rate (shown in Figure 3.38). Then, at the low end of the distribution, the number of transmission is too low to push the video throughput higher. In consequence, the video throughput drops significantly after the number of other mobile stations is over 25 (the 5MHz curve in Figure 3.36).

Voice bandwidth utilization in VSG-CDMA also has a tendency to drop under increasing load from non-voice traffic, unless W is very high. This tendency is shown in Figure 3.39. The wider the spread bandwidth, the more gradual the drop in voice bandwidth usage. In the 5MHz case, the voice bandwidth usage drops further when the number of other types of mobile stations is over 70. This mainly results from the increased multiple access interference level, which is mirrored by a further increase in the transmission failure rate, as shown in Figure 3.40

S-CDMA

The performance of S-CDMA under fixed load from a selected traffic class is very similar to what we have seen in Figure 3.18. The difference is that the bandwidth usage of the selected fixed-load service starts dropping from its maximum value, as shown in Figure 3.41.

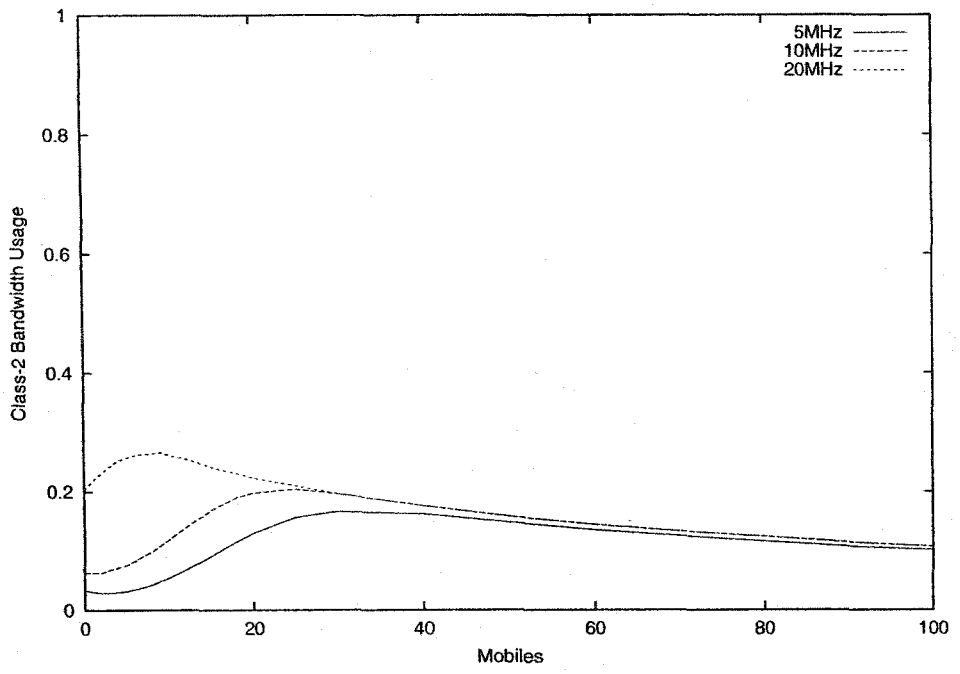
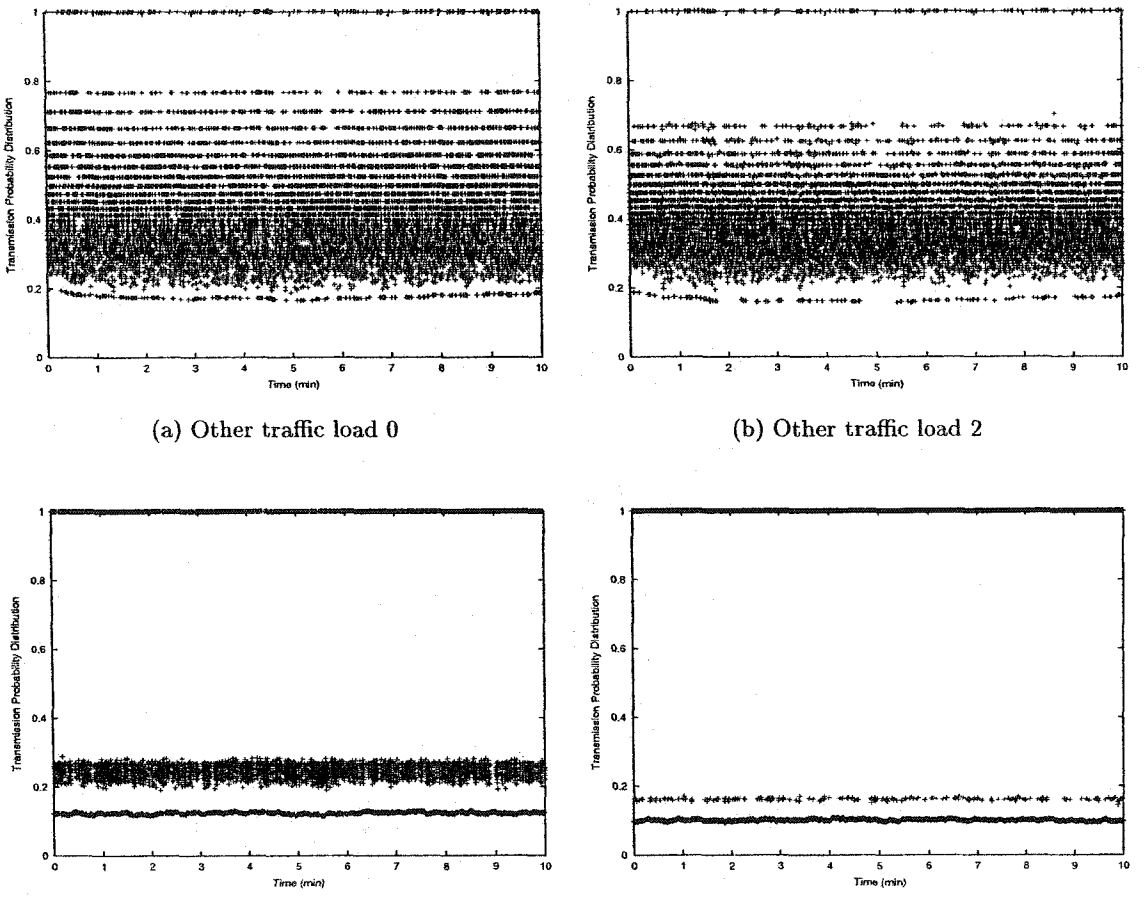


Figure 3.36: Fixed-load video bandwidth utilization in VSG-CDMA

3.6 Conclusions

We have presented a CDMA-based protocol aimed at accommodating traffic classes with different QoS requirements. By using the brick wall approach to partitioning the uplink frame among the multiple mobiles, the protocol is flexible with bandwidth allocation, yet the complexity of its bandwidth scheduler seems to be reasonably low. We have demonstrated that our protocol well caters to traffic classes with diverse QoS requirements and, in particular, efficiently accommodates data traffic without compromising the quality of service for voice and video. This property makes it a good candidate for future mobile networks, in which non-voice traffic will constitute a considerably more significant component than it does today.

At first sight, our approach of admitting only as many voice sessions as can be sustained simultaneously in their active phases may seem restrictive and run against the commonly accepted policy that relies on statistical multiplexing to offer more voice bandwidth to the users. Although one can only guess about the load patterns of future PCS networks, it is rather obvious that the contribution of traditional voice sessions to those patterns will tend to decrease with time. Consequently, with the increasing spread bandwidth of those networks, it will be pointless to try to accommodate as many voice sessions as physically possible, and the focus will shift toward efficient coexistence of voice with other session



(a) Other traffic load 0

(b) Other traffic load 2

(c) Other traffic load 25

(d) Other traffic load 50

Figure 3.37: Transmission probability distribution in VSG-CDMA ($W = 5\,\text{MHz}$, fixed video load)

types. In this context, one should not worry about the "holes" caused by inactive voice sessions, which will be naturally reused by less picky (but no less important) asynchronous transactions.

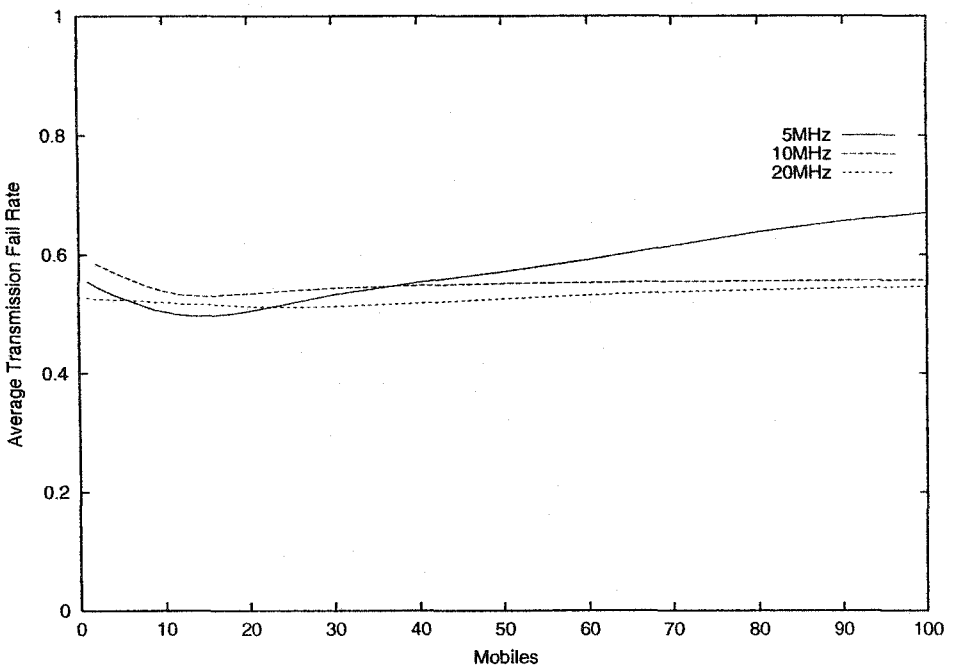


Figure 3.38: Transmission failure rate in VSG-CDMA under fixed video load

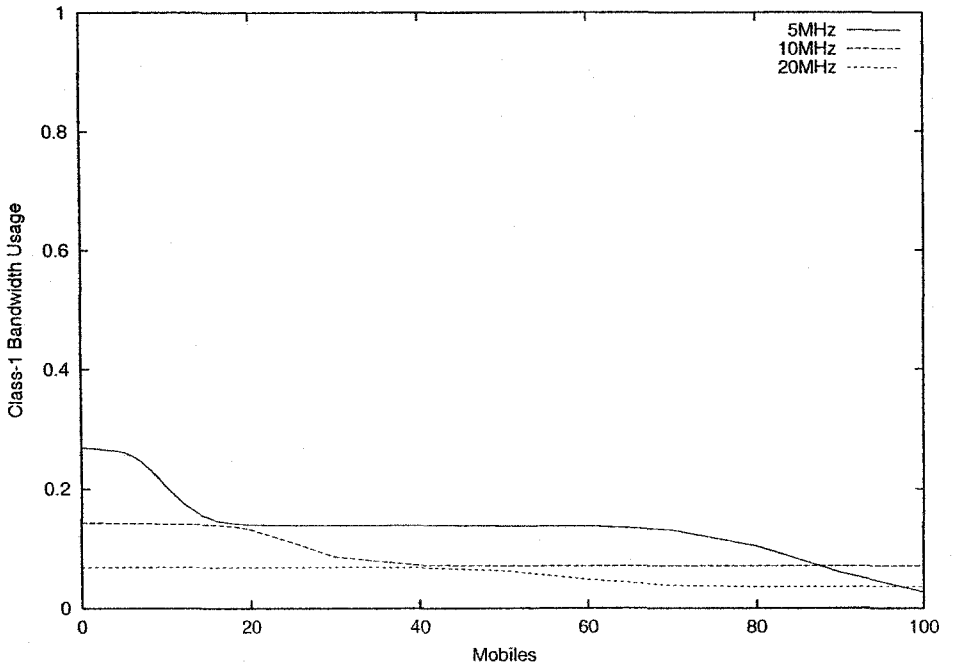


Figure 3.39: Fixed-load voice bandwidth utilization in VSG-CDMA

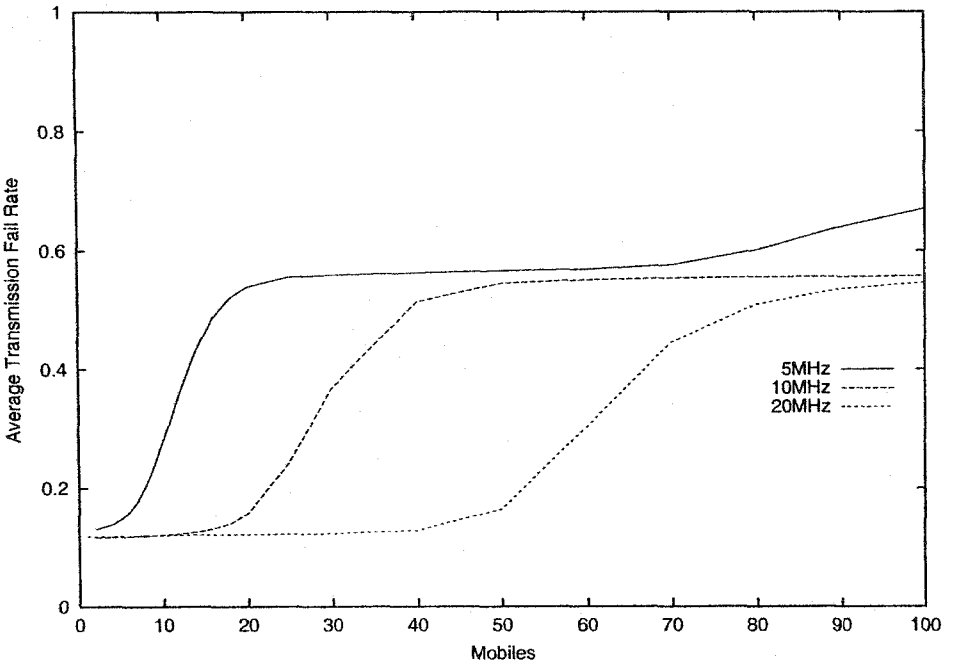


Figure 3.40: Transmission failure rate in VSG-CDMA under fixed voice load

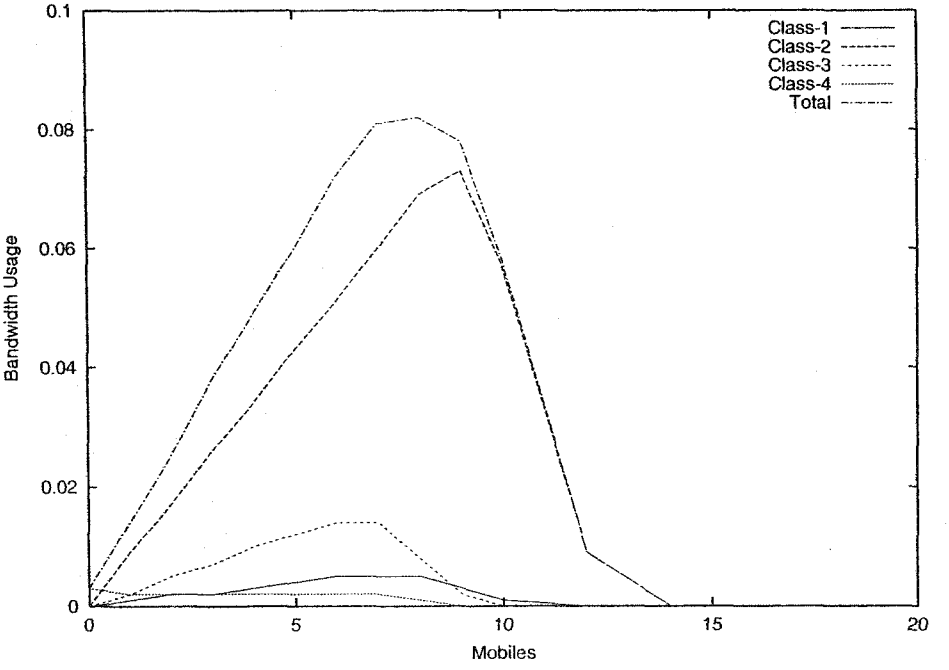


Figure 3.41: Fixed-load bandwidth utilization of SMS S-CDMA ($W = 20\,\text{MHz}$)

Chapter 4

Future Work

The future wireless communication systems will provide different services at the same time. Different applications have different bandwidth requirments. To use the bandwidth efficiently across the air interface, the MAC layer must provide a flexiable scheme to grant bandwidth. Our DS-TDMA/CP protocol meets this with the variable slot size design. Under this protocol, different bandwidth requirements are satisfied with with different slot sizes. Compared with fixed slot-size solutions, our solution saves bandwidth on framing, e.g. the bandwidth used on guard time. Also, this design makes the scheduling algorithms at base the station reuse the bandwidth of voice sessions in silent periods much easier. Our BRICS protocol meets the protocol flexibility requirement with the design of the physical radio channels and the brick wall approach on the bandwidth scheduling. Each radio channel is deisgned for one kind of bandwidth requirement. Compared with other CDMA solutions, our solution is able to accommodate traffic classes with different QoS requirements and saves bandwidth both from framing point of view and from transmitting power point of view. Also, a code acquisition system is proposed to improve our protocol's performance.

One problem with our protocols presented in this thesis, and with other proposed solutions that can be found in the literature, is the relatively high demand on CPU processing power at the base station. This seems to be a common drawback of the generic approach assuming that the base is solely responsible for bandwidth scheduling. Although, in our case, the complexity of the algorithms involved appears to be well within the reach of contemporary hardware, the exact magnitude of the processing power required at the base station and the dependence of this requirement on the parameters of the network should be investigated and quantified. The implementation of the scheduling algorithm at the base station, including the design of data structures used for storing the request queues, deserves some study as well.

A bandwidth-efficient medium access scheme necessarily involves a significant amount of signaling—to provide the requisite feedback both in the uplink and downlink directions. There is an obvious tradeoff between the amount of this feedback (that translates into the quality of bandwidth allocation) and the amount of bandwidth remaining for allocation (after the signaling bandwidth is deducted from the pool). On the other hand, the more resources spent on the signaling channels (i.e., bandwidth, MAI separation, power), the more reliable those channels become and, consequently, less of the allocatable bandwidth is misdirected, misused or destroyed, e.g., by interference. Putting those tradeoffs on a formal ground seems to be another interesting research direction.

The traditional multi-layered approach to organizing modern communication protocols has been extensively criticized as inadequate and wasteful, especially in the context of wireless networking. This issue is blatantly visible in modern cellular communication, where, for example, power control (rightfully belonging to the physical layer) must be accessible at least to the MAC layer. Besides power control, the code acquisition algorithm is another critical component of a CDMA scheme; the quality of this component has a direct and paramount impact on the bit error rate and the amount of bandwidth effectively available to the users. Traditionally, code acquisition schemes have been discussed as elements belonging strictly to the physical layer—in isolation from their role in the overall access protocol. As we point out at the end of Section 3.2.2, the code acquisition algorithm can be made more efficient if it receives some sensible feedback from the upper layers (notably the bandwidth scheduler) regarding the current "expected" uncertainty of the slot boundary. Currently, we are investigating the idea of improving the quality of CDMA code acquisition by integrating this part with the MAC scheme.

Appendix A

Methodology and Tools

A.1 Methodology

As noted in Section 1.1, the wireless communication systems of the future will cater to a variety of multimedia services and applications with diverse bandwidth requirements and quality of service (QoS) expectations. The demands of bandwidth and flexibility in access to the medium posed by those services and applications are confronted with the inherently limited capacity and restrictive characteristics of radio channels. From the review in Section 1.3 we can see that, new protocols for personal mobile communications must assume that non-voice traffic is an integral and important part of contemporary mobile networking. They must account for the presence of several different classes of traffic with diverse quality of service (QoS) requirements, and make sure that those classes coexist within the framework of limited bandwidth and flexibility of the mobile environment.

The focus of our work is to design bandwidth-efficient medium access control protocols for wireless communication systems that support multimedia applications. The primary methodology for the performance studies needed in our research is simulation. Simulation has some disadvantages, e.g., its results may be inaccurate compared to the results from practical experiments. Although in principle there is no limit to the accuracy of simulation models, one common problem with simulation studies in telecommunication is the lack of accurate models of realistic traffic scenarios. This problem is particularly visible, if the system under study addresses anticipated demands of future applications whose exact profile cannot be known at present. But this is also when simulation proves more convenient and effective than practical experiments—it allows us to study systems that do not exist under loads of nonexistent applications executed by nonexistent users. Even if a physical model of such a system could be built in principle, its cost in terms of money, effort, and time would render such an idea completely outlandish.

The simulation experiments reported upon in this work deal with the lowest layers of the protocol stack, i.e., layer 1 and layer 2. Besides the simulation model of the investigated network and protocol, the experimental setup includes a traffic generator supplying the network with an artificial collection of flows intended to mimic typical real-life traffic sessions.

A.2 Simulation tools

Computer simulation has been used in telecommunications research for a very long time. The list of simulation packages popular within the academic community includes NS2 [49] from the University of California, Berkeley, SIDE [50] from the University of Alberta, GloMoSim [54] from the University of California at Los Angeles, SIRCIM [52] from Virginia Tech, and REAL [51] from Cornell University. NS2 is a discrete event simulator targeted at networking research. The package provides substantial support for the simulation of TCP, routing and multicast protocols over wired and wireless (local and satellite) networks. In the MAC layer, CSMA/CD (Ethernet) and CSMA/CA (WaveLAN) are implemented. However, no ready tools are provided for modeling MAC protocols used in the cellular infrastructure. Regarding the physical layer, only the free space and two-ray propagation models are built into NS2.

GloMoSim is a collection of parallelized building blocks for modeling mobility, radio propagation, and wireless network protocols (from the MAC layer up to TCP). The built-in models of raw radio channels include free space, Rayleigh and Rician fading, but the log-normal model is not included. The simulator can only be used to model TDMA-based MAC protocols.

SIRCIM (Simulation of Indoor Radio Channel Impulse Response Models with Impulse Noise) was developed at MPRG based on years of studies involving measurements and modeling in a wide range of indoor, outdoor, cellular and micro-cellular environments. The package, which covers a wide selection of frequency bands, channel types, and many wireless applications, is focused on the physical layer.

REAL is a network simulator intended for studying the dynamic behavior of flow and congestion control schemes in packet-switched data networks.

The most famous commercial network simulation tool is OPNET [53]. However, this package focuses on modeling the existing commercially available solutions, rather than providing a research platform for devising novel protocols and networking concepts. In its radio/wireless Models, OPNET includes only AMPS and WLAN (IEEE 802.11).

Although some of the packages mentioned above offer rudimentary tools toward modeling raw physical radio channels and the MAC layer in wireless cellular communications, those tools are either very simple, or they focus on some narrow aspects, e.g., as in SIRCIM. Considering that radio bandwidth is perhaps the most scarce and precious resource in a wireless communications system, the MAC protocol acting as the manager for the radio bandwidth plays a very important role in the overall scheme. Since the MAC protocol directly operates on the physical layer, the best simulation package for research in wireless MAC protocols should include reasonably accurate and flexible models of the physical layer and the MAC layer. None of the above simulators comes close to fulfilling this postulate.

SIDE is a very good software toolkit. It provides a methodology for implementing control processes as collections of interacting, event-driven, dynamic objects. It offers a programming language for specifying and implementing the controllers of reactive systems, and has a run-time kernel for executing programs expressed in the SIDE specification language. It also provides a simulation engine for modeling systems specified in SIDE in a virtual but highly realistic environment, and utilizes a built-in reactive interface to the Internet. It is equipped with observers—tools for design verification and testing. Unfortunately, it is aimed exclusively at modeling wired networks and provides absolutely no built-in tools for simulating radio channels.

A.3 Traffic models

In order to study the performance of MAC protocols under load conditions resembling those of real-life diversified sessions, three generic traffic models have been implemented. They are the On-Off bursty model, the discrete autoregressive model and the long range dependent (or self-similar) model.

A.3.1 The On-Off model

The On-Off traffic model is the most commonly used model of voice traffic. It reflects the natural structure of the speech signal, which is either in the *talking* or *silent* mode. By assuming that a voice activity detector is used at the mobile, the station generates bursts of packets at a constant rate (CBR) while in *talking* mode, and no packets at all while in *silent* mode. Following the studies in [17, 55], our model assumes that the time spent in each state is exponentially distributed with two means: *off_mean* for the *Off* state and *on_mean* for the *On* state. As suggested in [56], a Poisson arrival process for new voice calls is assumed, and the duration of a single session is exponentially distributed as well. Our

description of a voice traffic can be viewed as a C++ class with the following interface:

```
class On_Off {
        RandEXP *on_exp, *off_exp;
        double on_mean, off_mean;
public:
        On_Off(double *data, int canon, long *seed);
        ~On_Off();
        double onInterval();
        double getOnMean();
        double offInterval()
        double getOffMean();
};
```

where *on_exp* and *off_exp* generate random numbers according to the exponential distribution; *on_mean* and *off_mean* are the mean (in second) for the active period and silent period; *data[0]* and *data[1]* are the initialization values for *on_mean* and *off_mean* respectively; *canon* is a random level between 0 and 10 (the higher the random level the longer the random sequence cycle); *seed* is an array that stores the random generator seeds; *onInterval()* generates a random *On* time interval (in second); *offInterval()* generates a random *Off* time interval (in second); *getOnMean()* and *getOffMean()* return the values of *on_mean* and *off_mean* respectively.

A.3.2 The discrete autoregressive model (1)

We adopt the DAR(1) (Discrete AutoRegressive model) to describe VBR video traffic. With this model, the VBR traffic is characterized by three parameters: the mean, the variance, and the first-order auto-correlation coefficient. It was found that the DAR(1) model provides sufficient accuracy in characterizing generic variable bit rate video traffic sources [57, 58]. Consequently, the DAR(1) model has been used in many simulation studies [59, 18]. As stated in [57, 59], the number of ATM cells per video frame follows the Gamma distribution (it can also be Pareto or Weibull distribution [60]). We used Gamma distribution in our simulator.

The DAR(1) process is a first order discrete-time Markov chain. The transition matrix is computed from

$$P = \rho I + (1 - \rho)Q \tag{A.1}$$

where ρ is the auto-correlation coefficient for the size of video frames, I is the identity matrix, and each row of Q consists of the negative binomial probabilities $(f_0, f_1, \cdots, f_K, F_K)$, where $F_K = \sum_{k>K} f_k$ and K is the peak rate. The probability function for the negative binormal distribution is given by

$$f_k = \begin{pmatrix} k+r-1 \\ k \end{pmatrix} p^r q^k = \begin{pmatrix} -r \\ k \end{pmatrix} p^r (-q)^k \qquad k = 0, 1, \cdots \tag{A.2}$$

Here, $0 < p < 1, q = 1 - p$ and $r > 0$. p and r are give by

$$p = \frac{m}{v} \qquad \text{and} \qquad r = \frac{mp}{1-p} \tag{A.3}$$

where m and v are the mean and the variance.

Our implementation of the DAR traffic model can be illustrated as a C++ class with the following interface:

```
class DAR1 {
        RandALFG *uni_rand;
        double mean_val, variance_val, correlation_val, *nbp;
        int status, state;
public:
        DAR1(double *data, int canon, long seed);
        ~DAR1();
        double negBinomialProb(double r_var, int k_var, double prob);
        int VBRRate();
        double getVBRParameters(double *, double *, double *);
};
```

where *uni_rand* is an uniform random number generator; *mean_val* is the mean of video frame size, which is implemented as the mean traffic rate (in kbps) in the simulator; *variance_val* and *correlation_val* are the variance and auto-correlation coefficient for the video frame size; *nbp* is an array that stores the negative binormal probabilities; *status* and *state* are the total number of Markov states and the state index; *data[0], data[1], data[2]* and *data[3]* are initialization values for *mean_val, variance_val, correlation_val* and *status* respectively; *cannon* has the same mean as in On_Off traffic model; *negBinomialProb()* is a negative binormal random number generator, and *r_var, k_var*, and *prob* are the r, k and P parameters in equation A.3 and A.2; *VBRRate()* generates the next traffic rate index; *getVBRParameter()* returns the DAR initialization parameters.

A.3.3 The self-similar traffic model

The self-similar traffic model [61] is the most well-known traffic model for Local Area Networks (LANs). One of its most striking features is the tremendous burstiness perceptible at practically any time scale, which is characteristic of a long-range dependent process. A self-similar process remains "bursty" under different levels of aggregation, i.e., a sum of many self-similar processes shows a behavior similar to each of the processes viewed individually. On the other hand, a sum of non self-similar processes tends to become smoother with aggregation and eventually assumes the characteristics of a Gaussian white noise. For our experiments, we implemented Hosking's algorithm [62, 63] to generate a long-range dependent process called fractional ARIMA(0, d, 0) and used that process to generate self-similar traffic. The corresponding C++ class has the following interface:

```
class LRDTraffic {
        char *trace_file;
        int mode_val, fd, fd_pipe[2], d_len;
        long pkgno_val;
        pid_t child_PID;
        double LRD_process;
        double H_val, d_val, v0_val;
public:
        LRDTraffic(int mode, double v0, double hurst, long pkg, char *tf);
        ~LRDTraffic();
        void getParameters(int *, double *, double *, long *, char *);
        double nextPoint();
        long generateTrace(long count);
};
```

where *trace_file* stores the trace file name; *mode_val* controls the object function mode: reading the trace from a file or generating the trace on-the-fly; *H_val* is the Hurst parameter; *d_val* is the *d* parameter in ARIMA(0, d, 0) model; *vo_val* is a normal distribution parameter in Hosking's algorithm; *child_PID* is the process ID of the UNIX child process that generates the LRD string; *fd, fd_pipe, d_len* are internal variables used for communication between processes; *pkgno_val* is the maximum packet sequence number; *LRD_process* is the next point variable in the LRD process; *getParameters()* returns the initialization parameters; *nextPoint()* generates the next point in the LRD process; *generateTrace(long*

count) generates a LRD trace, and *count* is the number of data in the trace.

A.4 Simulating the radio channel

This part of our experimental platform takes care of modeling the behavior of a generic raw wireless channel. The following classes of channels have been taken into account: Free Space Channel, Slow Fading Channel, Rayleigh Fading Channel, Rician Fading Channel, and Nakagami/Rician Fading Channel. Behavior models of those channels are relevant from the viewpoint of signal propagation and power control.

A.4.1 Free space propagation

The free-space propagation model [64, 65] provides a simple deterministic analysis of radio propagation. Despite its simplicity, this model does offer some insight into the basic propagation mechanisms and establish certain important bounds.

If a signal is transmitted from an isotropic lossless unity-gain antenna in free space, the transmitted energy spreads out uniformly in all directions. At a distance d, the transmission loss [64] defined as the ratio of transmitted power to received power (in decibels) on another isotropic lossless unity-gain receiving antenna is

$$L_{FS} = 32.44 + 20\log_{10}(f) + 20\log_{10}(d) \tag{A.4}$$

where L_{FS} is the free-space (FS) transmission loss in dB, f is the frequency in megahertz, and d is the distance in kilometers.

In our model, the free-space propagation channel is implemented as a C++ class with the following interface:

```
class FreeChannel {
        double loss;
        double f_val, d_val, pt_val, pr_val;
public:
        FreeChannel(double f, double pt);
        double setFrequency(double);
        double getFrequency();
        double setDistance(double);
        double getDistance();
        double setTransmittedPower(double);
```

```
        double getTransmittedPower();
        virtual double pathLoss();
        double signalPower();
};
```

where *loss*, *f_val*, and d_val are the L_{FS}, f and d in equation A.4; *pt_val*, and *pr_val* are the transmitted power (in dBm), and the received power (in dBm) respectively; *signalPower()* generates a signal level (in dBm) at the receiver; *setFrequency()* and *getFrequency()* are used to initialize and return the carrier frequency; *setDistance()* and *getDistance()* are used to initialize and return the distance parameter; *setTransmittedPower()* and *getTransmittedPower()* are used to initialize and return the transmission power parameter; *pathLoss()* calculates the path loss.

A.4.2 Slow fading

The slow fading model [65, 66, 67] describes the longer-term fading phenomenon, that is, the character of the average fading signal level over a small scale (on the order of tens of wavelengths). The longer-term variations in the local mean are caused by variations of shadowing and the diffraction of radio signals while the receiver is moving over distances large enough to produce significant variations in the terrain features between the transmitter and receiver. Because those variations are mostly caused by the mobile moving into the shadow of a hill or building, slow fading is often called shadowing. The local mean is a random variable itself due to the variations in the shadows. Empirical studies have shown that slow fading follows a log-normal distribution [67, P.87] with the PDF of the local mean described as

$$p(w) = \frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{(w-\mu)^2}{2\sigma^2}\right] \tag{A.5}$$

where w is the signal strength in dB, σ is the standard deviation in dB of the signal distribution, and μ is the mean signal level in dB.

The slow fading channel or shadow channel is implemented in our model as the following C++ class:

```
class ShadowChannel : public FreeChannel, public SlowFading {
        double mean, deviation;
public:
        ShadowChannel(double f, double pt, int canon, long seed);
        double setMeanLevel(double mu);
```

```
        double getMeanLevel();
        double setSTDDeviation(double sigma);
        double getSTDDeviation();
        double signalPower();
};
```

where *mean* and *deviation* are μ and σ in equation A.5; f and pt have the same mean as in free space propagation channel object; *canon* is a random level as in On-Off traffic object; *seed* is a seed for the random number generator; *setMeanLevel()* and *getMeanLevel()* are used to initialize and return the *mean* parameter; *setSTDDeviation()* and *getSTDDeviation()* are used to initialize and return the standard deviation parameter *deviation*; *signalPower()* generates the slow fading signal w in equation A.5.

A.4.3 Multipath fading - Rayleigh fading and Rician fading

Multipath fading [65, 66, 67] describes the short-term fading phenomenon. Due to the existence of multiple propagation paths between the transmitter and the receiver, the plane waves with different phases arrive at the receiver simultaneously. They combine vectorially at the receiver antenna causing constructive and destructive additions to the composite received signal, which manifest themselves as large variations in the amplitude and phase.

For a multipath fading channel containing no line-of-sight (LOS) path, the probability density function (PDF) of the signal envelope follows a Rayleigh distribution [65, P.120],

$$p(r) = \frac{r}{\alpha^2} exp\left[-\frac{r^2}{2\alpha^2}\right] \tag{A.6}$$

where $r^2/2$ is the short-term signal power [65, P.120], and α^2 is the mean power.

The Rayleigh fading model agrees very well with empirical observations for macrocellular environments. Rayleigh fading usually applies to any scenario in which there is no LOS path between the transmitter and receiver antennas. The Rayleigh fading channel is implemented in our simulator as the following C++ class:

```
class RayleighChannel : public ShadowChannel, public RayleighFading {
        double mean;
public:
        RayleighChannel(double f,double pt, int canon, long seed);
        double setAverage(double alpha, double mu, double sigma);
        double getAverage();
```

```
        double signalPower();
};
```

where *mean* is the average signal level (in dBm); f and pt have the same mean as in free space propagation channel object; *canon* is a random level as in On-Off traffic object; *seed* is a seed for the random number generator; *setAverage()* initializes the distribution parameters; *alpha* is the average signal level (in dBm) for Rayleigh fading; *mu* is the mean signal level (in dBm) for slow fading; *sigma* is the standard deviation (in dB) for slow fading; *getAverage()* returns the initialization parameters; *signalPower()* generates the Rayleigh fading signal (in dBm).

For a multipath fading channel containing a spectacular or LOS component, the probability density function (PDF) of the signal envelope follows a Rician distribution [65, P.135][67, P.47]. The Rician distribution is often described in terms of a parameter K [67, P.47] defined as

$$K = \frac{r_s^2}{2\sigma^2} \tag{A.7}$$

which is the ratio of the power in the steady (dominant) signal $r_s^2/2$ to that in the multipath (random) components σ^2. When $K = 0$, the channel exhibits Rayleigh fading, and when $K = \infty$, the channel exhibits no fading at all. For a Rician distributed envelope r (voltage), the average power is $\Omega_p/2 = r_s^2/2 + \sigma^2$ [65, P.120], and the Rician distribution can be written in terms of K as

$$p(r) = \frac{2r(K+1)}{\Omega_p} exp\left\{-K - \frac{(K+1)r^2}{\Omega_p}\right\} I_0\left(2r\sqrt{\frac{K(K+1)}{\Omega_p}}\right), \; r \geq 0 \tag{A.8}$$

where $I_0(.)$ is the modified Bessel function of the first kind and zero order. The Rician fading is very often observed in microcellular environments. Table A.1 [65, P.209] gives the optimum values of the parameter $K_{dB} = 10log_{10}(K)$, These values can be used as a starting point for estimating the channel characteristics and the performance of the radio system in specific classes of rural environments as a function of the transmission distance.

In our simulator, the Rician fading channel is implemented as the following C++ class:

```
class RicianChannel : public ShadowChannel, public RicianFading {
        double factor_k, mean;
public:
        RicianChannel(double f,double pt, int canon, long seed);
        double setAverage(double k,double h_omega,double mu,double sigma);
        double getAverage();
```

```
        double signalPower();
};
```

where *factor_k* is the K parameter (in dB) in equation A.8; *mean* (dBm) is a variable that represents the mean signal level at various circumstances; f and pt have the same mean as in free space propagation channel object; *canon* is a random level; *seed* is a seed for the random number generator; k is the Rice factor in dB; *h_omega* is the average signal level (in dBm) for Rician fading; *mu* is the mean signal level (in dBm) for slow fading; *sigma* is the standard deviation (in dB) for slow fading.

A.4.4 Nakagami fading

Over a relatively small distance (a few tens of wavelengths), signal propagation is well-described by Rayleigh or Rician statistics, with the local mean over a somewhat larger area (with homogeneous environmental characteristics) being lognormally distributed. Over large areas, the Nakagami distribution [65, P.153][67, P48] [68] is known to provide a closer match to experimental observations than the Rayleigh, Rician, or lognormal distributions [69].

In essence, the Nakagami distribution describes the envelope of the received radio signal envelope by a central chi-square distribution with m degrees of freedom,

$$p_z(r) = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega}\right)^m r^{2m-1} exp\left(-\frac{m}{\Omega} r^2\right) \quad m \geq \frac{1}{2} \tag{A.9}$$

where m and Ω are the parameters (Ω being the mean square value, $\Omega = 2\sigma^2$) and $\Gamma(\cdot)$ is the Gamma function. When $m = \frac{1}{2}$, the Nakagami distribution reduces to the one-sided Gaussian distribution, when $m = 1$, it becomes the Rayleigh distribution, and when $m \to \infty$, the distribution becomes an impulse (no fading). Thus, the Nakagami distribution can model fading conditions that are either more or less severe than Rayleigh fading. The Rician distribution can be closely approximated by Nakagami using the following relation

Terrain	Range≤6km	Range>6km
Woodland	-6.0	-14.0
Town	-1.2	-6.0
Village	0.6	-6.0
Hamlet	2.1	1.3
Rural lane	0.8	0
Minor road	0.9	0
Major road	1.4	-2.6

Table A.1: The optimum values of the Rice parameter K_{dB}

between the Rician factor K_{dB} and the Nakagami parameter m [68]:

$$K_{dB} = \frac{\sqrt{m^2 - m}}{m - \sqrt{m^2 - m}} \qquad m > 1 \tag{A.10}$$

$$m = \frac{(K_{dB} + 1)^2}{2K_{dB} + 1} \tag{A.11}$$

Since the Rician distribution contains a Bessel function, while the Nakagami distribution does not, the Nakagami distribution is easier transformable into closed form analytical expressions.

By using a transformation of random variables, the squared envelope $y = r^2$ has the Gamma density

$$p_{z^2}(y) = \left(\frac{m}{\Omega}\right)^m \frac{y^{m-1}}{\Gamma(m)} exp\left\{-\frac{my}{\Omega}\right\} \tag{A.12}$$

The Nakagami fading channel is implemented in our model as the following C++ class:

```
class NakagamiChannel : public ShadowChannel, public NakagamiFading {
        int ch_type;
        double degree_mk, mean;
        double (NakagamiChannel::* signal_power)();
public:
        NakagamiChannel(double f,double pt, int canon, long seed);
        int setChannelType(int type);
        int getChannelType();
        double setAverage(double mk,double h_omega,double mu,double sigma);
        double getAverage();
        double signalPower();
};
```

where *ch_type* indicates whether the channel is a Nakagami fading channel or a Rician fading channel; *degree_mk* represents the Rician factor K or the Nakagami parameter m in equation A.11; *mean* has the same meaning as in the Rician fading channel; f and pt have the same mean as in free space propagation channel object; *canon* is a random level; *seed* is a seed for the random number generator; mk is the Rice factor in dB or the Nakagami freedom in degrees; *h_omega* is the average signal level for Nakagami/Rician fading in dBm; *mu* is the mean signal level for slow fading in dBm; *sigma* is the standard deviation for slow fading in dB.

A.5 Modeling the air interface

In our study, we focus on the medium access control layer of the cellular system, ignoring the handover and other issues related to the fixed infrastructure of the interconnected base stations. Under these conditions the system to be investigated can be simplified to a single cell. That is, we assume that the system consists of a single base station and a number of mobile stations, which are randomly distributed within the cell. All traffic in the network is carried between the base station and the mobile stations. The population of mobile stations may grow and shrink; their traffic patterns and bandwidth requirements may vary dynamically.

In a cellular wireless communications system, the downlink (i.e., the channel from the base station to the mobiles) is used solely by the base station to send to the mobile stations packets as well as control information required by the access protocol for the uplink channels. The downlink channel is used in a broadcast mode and is never subject to contention. The uplink channel (from the mobiles to the base) is shared among all the mobile stations in the cell. This involves contention, multiple access control and bandwidth allocation.

As part of our work, we have developed a simulator to study the behavior of a one-cell radio communications system. The simulator is time driven and it uses a fixed time advance interval. This approach makes perfect sense for modeling cellular systems in which all transmissions are organized into frames centrally orchestrated by the base station. Consequently, a natural time step for the model is usually provided by the length of the TDMA frame or, possibly, some other duration of a standard cycle used by the base station to convey its "announcements" to the mobiles.

In our implementation of this model, the communication between the mobile stations and the base is carried out through two information boards: the uplink information board and the downlink information board, as shown in Figure A.1.

The mobile stations post information, e.g., uplink signals and traffic data, to the uplink information board, and they retrieve information, e.g., radio resource assignment and medium access control signals, from the downlink information board. The base station posts information to the downlink information board and reads information from the uplink information board.

The simulator has been programmed in C++ under Linux and Solaris. It has five basic elements: the main function, the system creator (or builder), the base station model, the generic model of a mobile station, and the performance deliverer. Depending on the services

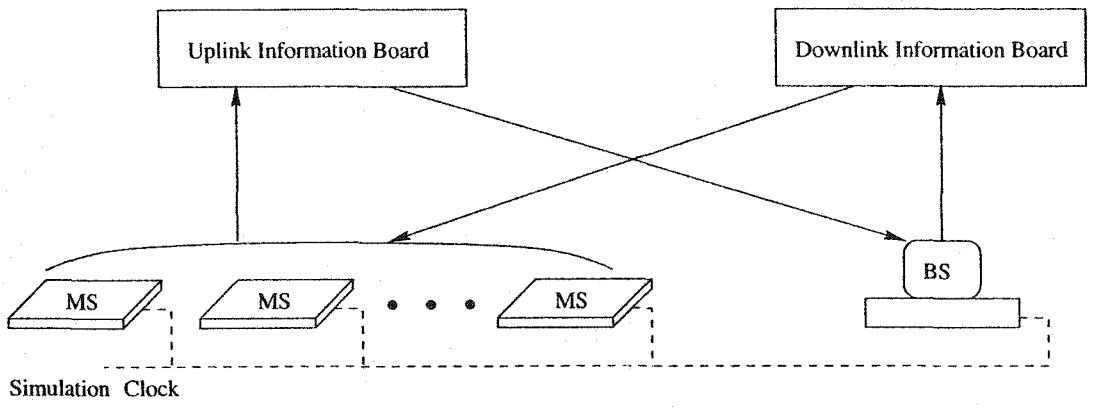


Figure A.1: The simulator structure

provided by the cellular system under study, different (actual) types of mobile stations are derived from the generic mobile station model. Each actual mobile station type represents one category of service.

A.5.1 The performance monitor

The performance monitor outputs the performance results from a simulation experiment to the standard output and/or a file. These results cover the following aspects:

- traffic parameters
- total throughput in bits or ATM cells (collected over all traffic classes, and individual throughput for each traffic class
- the amount of dropped traffic for each class and the calculated drop rate
- the average transmission delay for each class
- bandwidth utilization in percentage for each traffic class and others (if applicable), e.g. random access request, guard time, slot preamble, etc.
- access blocking rate, access delay, access request collision rate, etc. (for each traffic class)

The following C++ class shows a sample definition of performance monitor:

```
class Output {
        FILE *sd;
public:
```

```
        Output(FILE *des);
        void traffics(PERFORMANCE, MOBILESYSTEM);
        void throughput(PERFORMANCE, MOBILESYSTEM);
        void drop(PERFORMANCE, MOBILESYSTEM);
        void delay(PERFORMANCE);
        void utilization(PERFORMANCE);
        void callBlock(PERFORMANCE);
        void accessing(PERFORMANCE);
        void arts(PERFORMANCE, MOBILESYSTEM);
};
```

where *PERFORMANCE* is a C++ data structure that stores all performance results; *MOBILESYSTEM* is another C++ data structure that contains the base station and all mobile stations; *sd* is a file descriptor variable that represents *des*, which points to a file that stores the performance results; *void traffics(PERFORMANCE, MOBILESYSTEM)* outputs the traffic load of each traffic class to the standard output and to the indicated file; *void throughput(PERFORMANCE, MOBILESYSTEM)* outputs the throughput of each traffic class; *void drop(PERFORMANCE, MOBILESYSTEM)* outputs the amount of dropped traffic and the drop rate of every traffic class; *void delay(PERFORMANCE)* outputs the transmission delay for each traffic class; *void utilization(PERFORMANCE)* outputs the total bandwidth utilization and the bandwidth utilization for each traffic class; *void callBlock(PERFORMANCE)* outputs the access request blocking rate of each traffic class; *void accessing(PERFORMANCE)* outputs the access request collision rate and access delay for each traffic class; *void arts(PERFORMANCE, MOBILESYSTEM)* activates the above output functions.

A.5.2 The base station

The exact behavior of the base station depends on the protocol under which the base station operates. In the simplest case, it may just receive data from the mobile stations. It may also perform complicated tasks of allocating bandwidth to the mobiles, resolving contention among them, etc., which may require solving non-trivial optimization problems. Below we include a sample definition of a base station, which comes from the DS-TDMA/CP protocol discussed at length in chapter 2. This station maintains medium access request queues, makes transmission schedules (bandwidth allocation), and controls the mobile stations' access to the system.

```
class BaseStation {
        int MIN_MINISLOTS, MAX_MINISLOTS;
        int CBR_STATIONS, VBR_STATIONS, UBR_STATIONS;
        int CBR_S_OVERHEAD, VBR_S_OVERHEAD, UBR_S_OVERHEAD;
        int P_OVERHEAD, PAYLOAD, REQ_PACKET, ATMCELL;   /* bits */
        double channel_rate;
        SIM_TIME FRAME, GUARD_TIME, PA_TIME;
        SIM_TIME VBR_ASSIGNABLE, UBR_ASSIGNABLE;
        SIM_TIME MINISLOT_WIDTH, CBR_SH_TIME, VBR_SH_TIME, UBR_SH_TIME;
        SIM_TIME CBR_SLOT_WIDTH, CBR_MIN_SLOT_WIDTH, UBR_MIN_SLOT_WIDTH;
        Acc_Req_Que **req_que, *req_que_tail, *sch_que, *store;
        IB *post;
        PERFORMANCE *perform;
public:
        BaseStation(double *, IB *, PERFORMANCE *);
        ~BaseStation();
        void signalBoardInit();
        void moveTo(Acc_Req_Que **here, Acc_Req_Que **from,
                    Acc_Req_Que **pre, Acc_Req_Que **pointer);
        void insertReq(Acc_Req_Que *);
        void dropOverdue(SIM_TIME);
        void advance(SIM_TIME);
        int recvAccReq(SIM_TIME);
        SIM_TIME schedule(SIM_TIME);
        int scheduleCBR(SIM_TIME *, SIM_TIME *, int *);
        int scheduleVBR(SIM_TIME *, SIM_TIME *, int *);
        SIM_TIME scheduleUBR(SIM_TIME, SIM_TIME *);
        Acc_Req_Que *getStorage();
};
```

where *SIM_TIME* is a user-defined type representing moments or intervals of simulated time; *Acc_Req_Que* is a class representing the contents of an access request posed by a mobile station and representing the parameters of a traffic session; *IB* is a class describing the structure of a posting board (two such boards are used by the simulator—as described above) for exchanging information between the base station and the mo-

biles; *MIN_MINISLOTS* and *MAX_MINISLOTS* are the minimum and maximum numbers of minislots in the frame (see Section 2.4.2); *CBR_STATIONS, VBR_STATIONS,* and *UBR_STATIONS* are the number of mobile stations engaged in CBR, VBR and UBR sessions respectively; *CBR_S_OVERHEAD, VBR_S_OVERHEAD,* and *UBR_S_OVERHEAD* are slot overheads (in bits) for CBR, VBR, and UBR traffic classes; *P_OVERHEAD* is the packet overhead (in bits); *PAYLOAD* is the packet payload size (in bits); *REQ_PACKET* is the size of an access request packet (in bits); *ATMCELL* is the ATM cell size (bits); *channel_rate* is the transmission rate (in kbps); *FRAME* is the TDMA frame length (μs); *GUARD_TIME* is the guard time between slots (μs); *PA_TIME* is the slot preamble size (μs); *VBR_ASSIGNABLE* is the minimum allocatable unit of slot time (μs) for VBR traffic; *UBR_ASSIGNABLE* is the minimum allocatable unit of slot time (μs) for UBR traffic; *MINISLOT_WIDTH* is the minislot size (μs); *CBR_SH_TIME, VBR_SH_TIME,* and *UBR_SH_TIME* are the sizes of slot headers for CBR, VBR, and UBR traffic classes; *CBR_SLOT_WIDTH* and *CBR_MIN_SLOT_WIDTH* are the two sizes of a CBR slot: the standard one, allocated when the session is active (talkspurt), and the short one, assigned when the session is in its silent state (Section 2.3); *UBR_MIN_SLOT_WIDTH* is the minimum UBR slot size (μs); *req_que* and *req_que_tail* are the pointers to the front and tail of the access request queue; *sch_que* is the head pointer of the queue of those access requests that have been granted and scheduled by the base station; *store* is a pointer to the list of free request description items in memory; *post* holds the information boards; *perform* stores the performance results; *signalBoardInit()* initializes the information boards; *moveTo()* is used to move access requests between queues and update pointers related to those queues; *insertReq()* inserts an access request into its access request queue; *dropOverdue()* drops those access requests that have passed their deadlines; *advance()* advances the simulated time at the base station and controls the base station's operation; *recvAccReq()* collects access requests from the access request channels; *schedule(), scheduleCBR(), scheduleVBR(),* and *scheduleUBR()* are methods responsible for making scheduling decisions; *getStorage()* gets a memory block from the list of free list items pointed to by *store* or allocates a new memory block if the list is empty.

A.5.3 Generic mobile station

All specific types of mobile stations are derived from the generic mobile station class which declares the common attributes and functions for all mobile stations. Below we list the C++ class declaration of a sample generic mobile station. This C++ class was used in the

model of the DS-TDMA/CP protocol discussed in chapter 2.

```
class MobileStation {
        RandLCG transfer, ms_selector, req_decider;
public:
        int id, type, state, traffic_model, last_model, backoff;
        int req_ch, req_ch_try, req_ch_rec, this_minislots;
        int S_OVERHEAD, P_OVERHEAD, PAYLOAD, REQ_PACKET, ATMCELL;
        SIM_TIME FRAME, ACC_REQ_DD, slot_w, req_ch_dd, req_ch_timer;
        SIM_TIME idel_l, active_l, session_l;
        double channel_rate, load, backoff_p;
        DOWNIB_Tokens token_mask;
        IB *post;
        PERFORMANCE *perform;
        TRAFFIC traffic;
        LAST last;
        MobileStation(int *, long *, double *, IB *, PERFORMANCE *);
        ~MobileStation();
        int idel();
        int activeness();
        int requestChannel(SIM_TIME span);
        double accessBackoff();
};
```

where *transfer* is the random number generator that controls the station's transitions from idle to active; *ms_selector* is the random number generator used for contention resolution, i.e., randomized selection of the minislot for posing the access request (see Section 2.3); *req_decider* is the random number generator used for implementing the p-persistent behavior of the station in the backed-off state (Section 2.4.1), i.e., to determine if the station should pose its request in the next uplink frame; *id, type, state, traffic_model, last_mode* and *backoff* are the mobile's ID, service type, mobile active state, associated traffic model, active time model, and backoff indicator, respectively; *req_ch, req_ch_try, req_ch_rec,* and *this_minislots* describe the processing stage of the last request issued by the station to the base: *req_ch* indicates whether a request has been made or not, *req_ch_try* tells how many times a request has been made for the last backlogged session, *req_ch_rec* indicates if the base

station has received the request successfully, and *this_minislots* tells how many minislots are available in the next uplink frame; *S_OVERHEAD* is the slot overhead which includes modem preamble (PA) and slot header (SH) in DS-TDMA/CP as described in Section 2.3.1; *P_OVERHEAD, PAYLOAD, REQ_PACKET* and *ATMCELL* have the same meaning as in *BaseStation*; *FRAME* is the frame size; *ACC_REQ_DD* is the due date of the current access request; *slot_w* stores slot width requested by the mobile in the next frame; *req_ch_dd* and *req_ch_timer* are two variables for access request due date and access request count down timer respectively; *idel_l, active_l* and *session_l* are the mobile station's idle time, active time, and session time respectively; *load* is a threshold parameter that is used to control the number of active mobile stations; *backoff_p* is a backoff control variable; *token_mask* is the contents of the contention permission flags last received from the base (Section 2.3); *post* and *perform* have the same meaning as in *BaseStation*; *traffic* is the traffic model associated with this mobile station; *last* is the mobile station's active-time model; *idle()* counts down the mobile's idle period; *activeness()* performs initialization when the mobile station becomes active; *requestChannel()* sends an access request packet; *accessBackoff()* calculates the backoff parameter.

Depending on the service type, different types of mobile stations are derived from the generic C++ class *MobileStation*. For example, a CBR mobile is defines as follows:

```
class MobileCBR : public MobileStation {
        TRAFFIC_DATA buf;
        double traffic_rate, burst;
        SIM_TIME CBR_SLOT_WIDTH, CBR_MIN_SLOT_WIDTH;
public:
        MobileCBR(int *, long *, double *, IB *, PERFORMANCE *);
        ~MobileCBR();
        int requestStatus();
        int checkReqTimer();
        int talking();
        int silent();
        int advance(SIM_TIME sim_clock);
};
```

where *buf* stores CBR packets; *traffic_rate* is the CBR traffic rate; *burst* is a variable that represents the talking or silent period; *CBR_SLOT_WIDTH* and *CBR_MIN_SLOT_WIDTH*

have the same meaning as in *BaseStation*; *requestStatus()* checks the access request status broadcast by the base station; *checkReqTimer()* counts down the access request timer; *talking()* generates CBR traffic and transmits CBR packets to the base station; *silent()* counts down the silent period; *advance()* controls the station's activity in time according to its state, i.e., implements the state transition function based on the simulated time; *sim_clock* is a variable that indicates the current simulation time.

A.5.4 The system creator

The system creator reads the network and protocol parameters from the data file, initializes simulation objects and variables, and creates the base station and the mobile stations. The system creator is implemented as the following C++ class:

```
class Setup {
        int *model;
        long *rand_p;
        double *data;
        FILE *para, *sd;
        RandLCG *lcg_seed;
        MOBILESYSTEM one_cell;
        PERFORMANCE *perform;
public:
        Setup(FILE *des, PERFORMANCE *eval);
        ~Setup();
        void readData();
        void preprocessing();
        void conditions();
        void initializePerform();
        void createBase(IB *signal);
        void createCBR(IB *signal);
        void createVBR(IB *signal);
        void createUBR(IB *signal);
        void destroyMobiles();
        MOBILESYSTEM MobileNet(IB *signal);
};
```

where *model[0]* indicates the mobile non-idle time model, that is the time the mobile stays in active state; *model*[1] indicates the traffic model; *rand_p* stores seeds for random number generators; *data* stores the system and protocol parameters; *para* and *sd* are two file descriptors associated with the input and output files; *RandLCG* is a class type describing a linear congruential random number generator (LCG), and *lcg_seed* is an instance of such a generator; *MOBILESYSTEM* is a class type encompassing the simulated setup consisting of a base station, a number of mobile stations and the air interface interconnecting them into a communication system; *one_cell* is an instance of a mobile system system, i.e., the network to be simulated; *readData()* reads the system and protocol parameters from the input file; *preprocessing()* performs additional initialization, i.e., it calculates some system and protocol parameters based on input data; *conditions()* prints out the simulation parameters to the standard output; *initializePerform()* allocates memory for the *perform* class and initializes the variables in this structure; *createBase(), createCBR(), createVBR()* and *createUBR()* create the base station, the CBR mobiles, the VBR mobiles, and the UBR mobiles, respectively; *destroyMobiles()* releases the memory allocated to all stations; *MobileNet()* coordinates the functions within the system creator; *signal* holds the uplink and downlink signals.

A.5.5 The main function

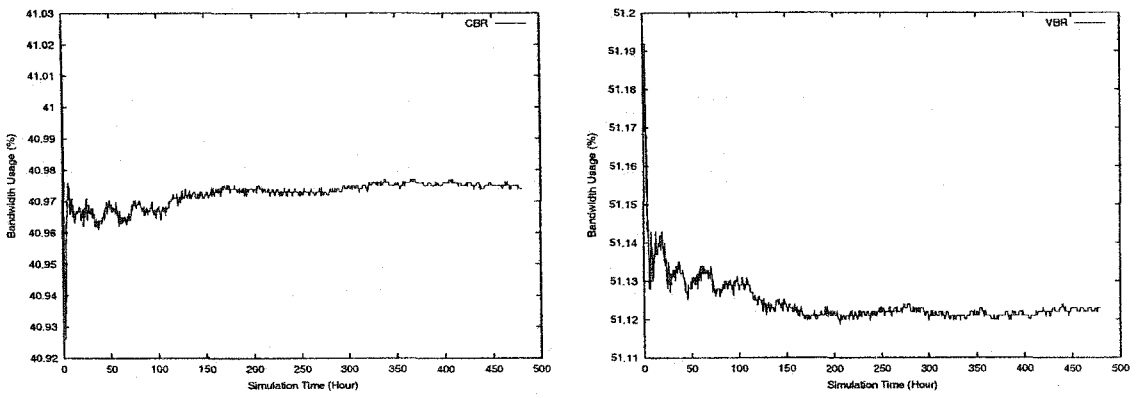
The simulator's main function controls the running sequence of the simulation experiment. It first sets the simulation environment for the mobile system and carries out the initialization, then it enters a loop to advance the simulated time and count down the simulation cycles. At each advance of the simulated time, the mobile stations are executed first, and they write to the uplink information board if applicable, e.g., send access request packets or data packets. Then, the base station is run. It reads the uplink information board, makes uplink access schedules, and broadcasts these schedules along with other signals via the downlink information board. Before the end of one simulation cycle, those mobiles that have issued access requests (either through explicit request packets or by piggybacking requests onto transmitted data packets—see Section 2.3) are run again to check the status of their requests. After the simulator completes all the prescribed cycles, it prints out the simulation results, frees the memory, and then exits.

The above discussion refers to the simulator for the DS-TDMA/CP protocol that will be discussed in the next chapter. This simulator is one of seven simulators that we built in the course of our study. Those other simulators implement models of the following protocols:

D-TDMA, DQRUMA, S-CDMA, VSG-CDMA, WISPER, and BRICS. Depending on the protocol requirements, more or less functionality was required with respect to the discussed DS-TDMA/CP simulator. For example, power control simulation was added in S-CDMA and VSG-CDMA models.

A.5.6 Simulation Methodology

We took two measures to ensure the correctness and fidelity of the simulation results. First, the observations were averaged over the whole simulation time, such as the bandwidth usage of a traffic class was an average percentage on the total available bandwidth of the simulation time. Second, we carried out some experiments to determine the simulation stabilization time. So the warm-up periods are averaged out at the end of the simulation and the simulation results can reach a relatively stable level. Figure A.2 shows the stabilization curves of CBR and VBR bandwidth utilization under DS-TDMA/CP. We can see that the



(a) CBR Bandwidth Usage

(b) VBR Bandwidth Usage

Figure A.2: Bandwidth Usage Stabilization Time under DS-TDMA/CP

simulation results become relatively stable after 160 hours of simulation time. Also, we can see that the randomness in the simulation results can not be averaged out completely.

The protocol performance experiments are carried out in the following manner. We set the simulation time to its stabilization time, e.g. 8 days, and set other protocol parameters. After the simulator advances 8-day simulation time, it will write the simulation results to a file and stop. If the X-axis is the number of mobile stations, we will change the number of mobile stations in the simulation configuration and repeat the simulation. If it needs 10 points to make the performance curve, 10 simulation experiments will be done.

Abbreviations

3G Third Generation

ABR Available Bit Rate

ACK Acknowledgment

AMPS Advanced Mobile Phone System

ARIMA AutoRegressive Integrative Moving Average

ATM Asychronous Transfer Mode

BER Bit Error Rate

CBR Constant Bit Rate

CDMA Code Division Multiple Access

CFU Call Forwarding Unconditional

CFNR Call Forwarding on mobile subscriber Not Reachable

CN Core Network

CM Connection Management

CPF Contention Permission Flags

C-PRMA Centralized Packet Reservation Multiple Access

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CSMA/CD Carrier Sense Multiple Access with Collision Detection

CUG Closed User Group

DAR Discrete Autoregressive Model

DCH Dedicated Channel

DPRMA Dynamic Packet Reservation Multiple Access

DQRUMA Distributed Queuing Request Update Multiple Access

DRMA Dynamic Reservation Multiple Access

DS-CDMA Direct Sequence Code Division Multiple Access

DSSS Direct Sequence Spread Spectrum

DS-TDMA/CP Dynamically Slotted TDMA with Contention Permission

D-TDMA Dynamic Time Division Multiple Access

ETSI European Telecommunications Standards Institute

FACH Forward Access Channel

FDD Frequency Division Duplex

FDMA Frequency Division Multiple Access

FHSS Frequency Hopping Spread Spectrum

FIFO First-In-First-Out

GSM Global System for Mobile Communications

GPRS General Packet Radio Service

HSD High Speed Data services

IMT International Mobile Telecommunication

IR Infrared

IS-95 Interim Standard 95

ISDN Integrated Services Digital Network

LAC Link Access Control

LAN Local Area Network

LAPDm Link Access Protocol for Data mobile channel

LAPD Link Access Protocol D channel

LCG Linear Congruential Generator

LEP Least Effort Policy

LOS Line Of Sight

MAC Medium Access Control

MAI Multiple Access Interference

MC-CDMA Multi Code Code Division Multiple Access

MCPA Maximum Capacity Power Allocation

MM Mobility Management

NACK Negative Acknowledgment

nrt-VBR non-real-time Variable Bit Rate

NS Network Simulator

PCS Personal Communication System

PDF Probability Density Function

PDU Protocol Data Unit

PN Pseudo-Noise

PRMA Packet Reservation Multiple Access

PTP Point-To-Point

QoS Quality of Service

OSI Open System Interconnection

RA Request Access

RAB Radio Bearer Service

RACH Random Access Channel

RAMA Resource Auction Multiple Access

RA-PN Random Access Pseudo-Noise code

RLC Radio Link Control

RR Radio Resources Management

RRC Radio Resources Control

rt-VBR real-time Variable Bit Rate

SDU Service Data Unit

SIDE Sensors In a Distributed Environment

SIR Signal to Interference Ratio

SIRCIM Simulation of Indoor Radio Channel Impulse Response Models

SMS Short Messaging Service

STE Shortest Time to Extinction

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

UBR Unspecified Bit Rate

UTRA UMTS Terrestrial Radio Access

UTRAN UMTS Terrestrial Radio Access Network

UMTS Universal Mobile Telecommunication System

VSG-CDMA Variable Spreading Gain CDMA

WISPER WIreleSs multimedia access control Protocol with [B]ER scheduling

WLAN WaveLAN/Wireless Local Area Network

WCDMA Wideband Code Division Multiple Access

Xmt Transmission

Bibliography

[1] H. Hammuda, Cellular Mobile Radio Systems: Designing Systems for Capacity Optimization. Chichester; New York: John Wiley & Sons. 1997.

[2] T. S. Rappaport, Wireless Communications: Principles and Practice. Prentice Hall, 1996.

[3] D. G. Jeong, W. S. Jeon, "A Data Transmission Scheme for CDMA Wireless Networks Based on IS-95," IEEE Transactions on Vehicular Technology, 49(1), pp. 11 -20, Jan. 2000.

[4] D.N. Knisely, S. Kumar, S. Laha, S. Nanda, "Evolution of Wireless Data Services: IS-95 to CDMA2000," IEEE Communications Magazine, Vol.36 No.10, pp.140 -149, Oct. 1998.

[5] C. Bettstetter, H.J. Vgel, and J. Eberspcher, "GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface," IEEE Communication Surveys, 2(3), 3rd Quarter 1999.

[6] T. Ojanpera, R. Prasad, "An Overview of Air Interface Multiple Access for IMT-2000/UMTS," IEEE Communications Magazine, 36(9), pp.82 -86, 91-95, Sept. 1998.

[7] F. Muratore, Mobile Communications for The Future. Chichester ; New York : Wiley, 2001.

[8] S. Dixit, Y. Guo, Z. Antoniou, "Resource Management and Quality of Service in Third Generation Wireless Networks," IEEE Communications Magazine, 39(2), pp.125 -133, Feb.2001.

[9] ATM Forum, "ATM Service Categories: The Benefits to The User," http://www.atmforum.com/.

[10] 3GPP home page: http://www.3GPP.org.

[11] M. Rahnema, "Overview Of The GSM System and Protocol Architecture," IEEE Communications Magazine, April 1993.

[12] E. Dahlman, P. Beming, J. Knutsson, F. Ovesjo, M. Persson, C. Roobol, "WCDMA-the Radio Interface for Future Mobile Multimedia Communications," IEEE Transactions on Vehicular Technology, 47(4), pp.1105-1118, Nov. 1998.

[13] A. Baier et al., "Design Study for a CDMA-Based Third-Generation Mobile Radio System," IEEE Journal on Selected Areas in Communications, Vol. 12, pp. 733-743, May 1994.

[14] A. J. Viterbi, "Wireless Digital Communication: A View Based on Three Lessons Learned," IEEE Communication Magazine, pp.33-36, Sept. 1991.

[15] M.W Oliphant, "Radio Interfaces Make The Difference in 3G Cellular Systems," IEEE Spectrum, 37(10), pp.53-58, Oct. 2000.

[16] F. Muratore, UMTS: Mobile Communications for The Future. Chichester ; New York : Wiley, 2001.

[17] D.J. Goodman, R.A. Valenzuela, K.T. Gayliard and B. Ramamoorthi, "Packet Reservation Multiple Access for Local Wireless Communications", IEEE Transactions on Communications, Vol. 37, No. 8, pp.885-890, August 1989.

[18] D.A. Dyson and Z.J. Haas, "A Dynamic Packet Reservation Multiple Access Scheme for Wireless ATM," In Proceedings of IEEE MILCOM'97, pp.687-693, Nov. 1997.

[19] G. Bianchi, F. Borgonovo, L. Fratta, L. Musumeci, M. Zorzi, "C-PRMA: a Centralized Packet Reservation Multiple Access for Local Wireless Communications", IEEE Transactions on Vehicular Technology, Vol. 46, pp. 422-436, May 1997.

[20] X. Qiu and V.O.K. Li, "Dynamic Reservation Multiple Access (DRMA): A New Multiple Access Protocol for Personal Communication Systems (PCS)," Wireless Networks, Vol. 2, no. 2, June 1996.

[21] G. Falk, J. Groff, W. Milliken, M. Nodine, S. Blumenthal, and W. Edmond, "Integration of Voice and Data in the Wideband Packet Satellite Network", IEEE Journal on Selected Areas in Communications, Vol.1, No. 6, December 1983.

[22] N.D. Wilson, R. Ganesh, K. Joseph, and D. Raychaudhuri, "Packet CDMA Versus Dynamic TDMA for Multiple Access in An Integrated Voice/Data PCN." IEEE Journal on Selected Areas in Communications, Vol. 11, pp.870-884, August 1993.

[23] N. Amitay, "Distributed Switching and Control with Fast Resource Assignment/Handoff for Personal Communication Systems", IEEE Journal on Selected Areas in Communications, Vol.11, pp.842-849, 1993.

[24] M. J. Karol, Z. Liu and K. Y. Eng, "Distributed-Queuing Request Update Multiple Access (DQRUMA) for Wireless Packet (ATM) Networks," In Proceeding of IEEE International Communications Conference, pp.1224-1231, June 1995.

[25] L. Kleinrock and H. Levy, "On the Behavior of A Very Fast Bidirectional Bus Network," In Proceedings of the 1987 IEEE International Communications Conference, pp.1419-1426, Seattle, Washington, June 1987.

[26] J. Capetanakis, "Tree Algorithms for Packet Broadcast Channels," IEEE Transactions on Information Theory, Vol.25, pp.505-515, 1979.

[27] J. F. Frigon, V.C.M. Leung, H. C. B. Chan, "Dynamic Reservation TDMA Protocol for Wireless ATM Networks," IEEE Journal on Selected Areas in Communications, 19(2), pp.370 -383, Feb 2001.

[28] T.V.J. Ganesh Babu, T. Le-Ngoc, J.F. Hayes, "Performance of A Priority-Based Dynamic Capacity Allocation Scheme for Wireless ATM Systems," IEEE Journal on Selected Areas in Communications, 19(2), pp.355-369, Feb 2001.

[29] M.C. Yuang, P.L Tien, "Multiple Access Control with Intelligent Bandwidth Allocation for Wireless ATM Networks," IEEE Journal on Selected Areas in Communications, Vol.18 No.9 , pp.1658 -1669, Sept. 2000.

[30] C. G. Kang, C. W. Ahn, K. H. Jang, W. S. Kang, "Contention-Free Distributed Dynamic Reservation MAC Protocol with Deterministic Scheduling (C-FD/sup3/R MAC) for Wireless ATM Networks," IEEE Journal on Selected Areas in Communications, 18(9), pp.1623 -1635, Sept. 2000.

[31] Y. K. Kwok, V.K.N.Lau, "A Quantitative Comparison of Multiple Access Control Protocols for Wireless ATM," IEEE Transactions on Vehicular Technology, 50(3), pp.796 -815, May 2001.

[32] D. Raychaudhuri, "Performance Analysis of Random Access Packet Switched Code Division Multiple Access Systems," IEEE Transactions on Communications, Vol. COM-29, pp. 895-901, Jun 1981.

[33] Z. Liu and M. E. Zarki, "Performance Analysis of DS-CDMA with Slotted ALOHA Random Access for Packet PCNs", Wireless Networks 1(1):1-16, Feb. 1995.

[34] R.K. Morrow and J.S.Lehnert,"Packet Throughput in Slotted ALOHA DS/SSMA Radio Systems with Random Signature Sequences,"IEEE Transactions on Communications, Vol.40, no.7, pp.1223–1230, July 1992.

[35] O. Sallent, R. Agusti, "A Proposal for An Adaptive S-ALOHA Access System for A Mobile CDMA Environment", IEEE Transactions on Vehicular Technology, Vol. 47 No. 3 , pp.977 -986, Aug. 1998.

[36] C. L. I and R.D Gitlin, "Multi-code CDMA Wireless Personal Communication Networks," in Proceedings of the IEEE International Conference on Communications, Seattle, WA, vol. 2, pp. 1060–1064, June 1995.

[37] C. L. I, K.K. Sabnani, "Variable Spreading Gain CDMA with Adaptive Control for True Packet Switching Wireless Network," in proceeding of ICC '95 Seattle, Vol.2, pp.725 -730, June 1995.

[38] R. Fantacci, S. Nannicini, "Multiple Access Protocol for Integration of Variable Bit Rate Multimedia Traffic in UMTS/IMT-2000 Based on Wideband CDMA," IEEE Journal on Selected Areas in Communications, Vol.18, No.8, pp.1441 -1454, Aug. 2000.

[39] S. J. Oh, K.M. Wasserman, "Dynamic Spreading Gain Control in Multiservice CDMA Networks," IEEE Journal on Selected Areas in Communications, Vol.17, No.5, pp.918 -927, May 1999.

[40] I. F. Akyildiz, D. A. Levine, and I. Joe, "A Slotted CDMA Protocol with BER Scheduling for Wireless Multimedia Networks", IEEE/ACM Transactions on Networking, Vol.7, pp.146 - 158, Apr. 1999.

[41] Z. Liu, M. J. Karol, M. E. Zarki and K. Y. Eng, "Channel Access and Interference Issues in Multi-Code DS-CDMA Wireless Packet (ATM) Networks", Wireless Networks, Vol. 2, pp.173-193, 1996.

[42] S. Choi and K. G. Shin, "An Uplink CDMA System Architecture with Diverse QoS Guarantees for Heterogeneous Traffic", IEEE/ACM Transactions on Networking, Vol.7, pp. 616 - 628, Oct. 1999.

[43] S. Kumar, S. Nanda, "High Data-Rate Packet Communications for Cellular Networks Using CDMA: Algorithms and Performance," IEEE Journal on Selected Areas in Communications, Vol.17, No.3, pp.472 -492, March 1999.

[44] J. M. Capone and L. S. Merakos, "Integrating Data Traffic into a CDMA Cellular Voice System," Wireless Networks, Vol.1, pp.389 - 401, Feb. 1995.

[45] A. Sampath, J.M. Holtzman, "Access Control of Data in Integrated Voice/Data CDMA Systems: Benefits and Tradeoffs," IEEE Journal on Selected Areas in Communications, Vol.15 No.8, pp.1511 -1526, Oct. 1997.

[46] T. K. Liu, J.A. Silvester, "Joint Admission/Congestion Control for Wireless CDMA Systems Supporting Integrated Services," IEEE Journal on Selected Areas in Communications, Vol.16, No.6, pp.845 -857, Aug. 1998.

[47] C. L. I, S. Nanda, "Load and Interference Based Demand Assignment (LIDA) for Integrated Services in CDMA Wireless Systems," in Proc. IEEE GLOBECOM, Vol.1, pp.235 -241, 1996.

[48] S. Ramakrishna, J.M. Holtzman, "A Scheme for Throughput Maximization in a Dual-Class CDMA System," IEEE Journal on Selected Areas in Communications, Vol. 16 No. 6, pp. 830 -844, Aug. 1998.

[49] "The Network Simulator - NS-2", http://www.isi.edu/nsnam/ns

[50] "SIDE", http://www.cs.ualberta.ca/ pawel/SIDE/

[51] "REAL 5.0 Overview", http://www.cs.cornell.edu/skeshav/real/overview.html

[52] "Simulation of Indoor Radio Channel Impulse Response Models with Impulse Noise (SIRCIM)", http://www.mprg.ee.vt.edu/research/sircim/sircim.html

[53] "OPNET", http://www.opnet.com/

[54] http://www.cs.ucla.edu/NRL/wireless/index.html

[55] X. Qiu, V. O. K. Li, and J. H. Ju, "A Multiple Access Scheme for Multimedia Traffic in Wireless ATM", Mob. Netw. Appl. 1(3), pp. 259 - 272, Dec. 1996.

[56] J. G. Markoulidakis, G. L. Lyberopoulos and M. E. Anagnostou, "Traffic Model for Third Generation Cellular Mobile Telecommunication Systems", Wireless Networks 4(5), pp. 389 - 400, Aug. 1998.

[57] D. P. Heyman, A. Tabatabei, and T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks," IEEE Transactions on Circuits and Systems for Video Technology, vol. 2, pp. 49–59, Mar. 1992.

[58] D. Heyman, T.V. Lakshman, A. Tabatabai, H. Heeke,"Modeling Teleconference Traffic from VBR Video Coders", IEEE International Conference on Communications, vol.3, pp. 1744 - 1748, May 1994.

[59] J. C. Chen, K. M. Sivalingam and R. Acharya, "Comparative Analysis of Wireless ATM Channel Access Protocols Supporting Multimedia Traffic", Mob. Netw. Appl. 3(3), pp. 293 - 306, Sep. 1998.

[60] D. P. Heyman and T. V. Lakshman, "Source Models for VBR Broadcast-Video Traffic", IEEE/ACM Trans. Networking 4(1), pp. 40 - 48, Feb. 1996.

[61] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)", IEEE/ACM Trans. Networking 2(1), pp. 1 - 15, Feb. 1994.

[62] J. R. M. Hosking, "Modeling Persistence in Hydrological Time Series Using Fractional Differencing." Water Resources Research, 20(12): pp. 1898-1908, 1984.

[63] M. W. Garrett, and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic", Proceedings of the Conference on Communications Architectures, Protocols and Applications, pp. 269 - 280, 1994.

[64] A. A. Smith, Radio Frequency Principles and Applications. New York, NY: IEEE Press, 1996

[65] D. Parsons, The Mobile Radio Propagation Channel. New York-Toronto: Halsted Press, 1992

[66] W. C.Y. Lee, Mobile Communications Design Fundamentals, Second Edition. New York, NY: John Wiley & Sons, 1993

[67] G. L. Stuber, Principles of Mobile Communication. Norwell, MA: Kluwer Academic Publishers, 1996

[68] M. Nakagami, "The m Distribution; a General Formula of Intensity Distribution of Rapid Fading", Statistical Methods in Radio Wave Propagation, W.G. Hoffman, ed., pp.3-36, 1960

[69] U. Charash, "Reception through Nakagami Fading Multipath Channels with Random Delays", IEEE Transactions on Communications, Vol.27, pp.657-670, April 1979.

[70] K.S. Gilhousen, I.M. Jacobs, R. Padovani, A.J. Viterbi, L.A. Weaver, Jr., and C.E. Wheatley III, "On the Capacity of a Cellular CDMA System," IEEE Transactions on Vehicular Technology, Vol.40, pp.303 - 312, May 1991.

[71] S. Choi and K. G. Shin, "A Cellular Wireless Local Area network with QoS Guarantees for Heterogeneous Traffic," Mobile Networks and Applications, 3(1), pp.89 - 100, June 1998.

[72] S. S. Panwar, D. Towsley and J. K. Wolf, "Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service," Journal of the ACM, Vol.35, pp.832-844, Oct. 1988.

[73] E. Sourour, S.C. Gupta, "Direct-Sequence Spread-Spectrum Parallel Acquisition in Nonselective and Frequency-Selective Rician Fading Channels," IEEE Journal on Selected Areas in Communications, Vol.10, No.3, pp.535 -544, April 1992.

[74] U. Madhow, M.B. Pursley, "Mathematical Modeling and Performance Analysis for a Two-Stage Acquisition Scheme for Direct-Sequence Spread-Spectrum CDMA," IEEE Transactions on Communications, Vol.43 No.9, pp.2511 -2520, Sept. 1995.

[75] H. R. Park, B. J. Kang, "On the Performance of a Maximum-Likelihood Code-Acquisition Technique for Preamble Search in a CDMA Reverse Link," IEEE Transactions on Vehicular Technology, Vol.47, No.1, pp.65 -74, Feb. 1998.

[76] A. Polydoros and C.L. Weber, "A Unified Approach to Serial Search Spread-Spectrum Code Acquisition - Parts II: A Matched-Filter Receiver," IEEE Transactions on Communications, Vol.32, No.5, pp.550-560, 1984.

[77] L. Fanucci, R. de Gaudenzi, F. Giannetti, M. Luise, "VLSI Implementation of a Signal Recognition and Code Acquisition Algorithm for CDMA Packet Receivers," IEEE Journal on Selected Areas in Communications, Vol.16 No.9, pp.1796 -1808, Dec. 1998.

[78] U. Madhow and M. B. Pursley, "Acquisition in Direct-Sequence Spread-Spectrum Communication Networks: an Asymptotic Analysis," IEEE Transactions on Information Theory, Vol.39, No.3, pp. 903-912, May 1993.

[79] A.M. Viterbi, and A.J. Viterbi, "Erlang Capacity of a Power Controlled Cellular CDMA System," IEEE Journal on Selected Areas in Communications, Vol.11, No 6, pp.892-900, August, 1993.

[80] L. C. Yun and D. G. Messerschmitt, "Power Control for Variable QOS on a CDMA Channel," In Proceedings of IEEE MILCOM, Vol.1, pp.178-182, 1994.

[81] A. Sampath, P. S. Kumar and J. M. Holtzman. "Power Control and Resource Management for a Multimedia Wireless CDMA System". In PIMRC'95, September 1995.

[82] P. Gburzynski, P. Rudnicki, "A Virtual Token Protocol for Bus-Type Networks: Correctness and Performance," Infor, Vol.26, pp.365 -393, 1988.

[83] X. Qiu, V. O. K. Li, and J. H. Ju, "A Multiple Access Scheme for Multimedia Traffic in Wireless ATM," Mobile Networks and Applications, 1(3):259 - 272, Dec. 1996.

[84] S. Verdu, "Minimum Probability of Error for Asynchronous Gaussian Multiple-access Channels, " IEEE Transactions on Information Theory, vol. IT-32, no.1, pp.85 - 96, 1986.

[85] T. J. Lim, L.K. Rasmussen, H. Sugimoto, "An Asynchronous Multiuser CDMA Detector Based on the Kalman Filter," IEEE Journal on Selected Areas in Communications, Vol.16, No.9, pp.1711 - 1722, Dec. 1998.