MULTIMEDIA SUPPORT FOR WIRELESS CDMA WITH DYNAMIC SPREADING

By

MEHMET ALI ELICIN

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2001

Dedicated to my father Tuncer Elicin, to my mother K. Saba Doganlarli,

to my best friend Ozgur Ozen and

to the memory of my beloved sister Aysegul Elicin.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Jonathan Liu, for giving me an opportunity to work on this challenging topic and for providing continuous guidance and feedback during the course of this work and thesis writing. He has been a great source of inspiration and my mentor.

I wish to thank Dr. Randy Chow and Dr. Stephen Thebaut for serving on my supervisory committee and for their careful review of this thesis.

I express my gratitude to Ju Wang for his valuable suggestions, guidance, and support without which I would not have accomplished this work.

I also would like to take this opportunity to thank my parents and my friend Ozgur Ozen for their inspiration and emotional support throughout my academic career.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

MULTIMEDIA SUPPORT FOR WIRELESS CDMA
WITH DYNAMIC SPREADING

By

Mehmet Ali Elicin

May 2001

Chair: C. L. Jonathan Liu
Major Department: Computer and Information Science and Engineering

The emerging M-commerce needs more support in operating systems in order to be successful over a wireless environment. The system needs to support a seamless integration (i.e., transparent application switching) to support voice, audio and conventional data (e.g., e-mails, and ftp). It should also support many users with guaranteed quality. In this thesis, we investigate effective protocol design with dynamic spreading factors such that various QoS based on different traffic types can be provided. Increasing spreading factors can benefit the system because it will increase the desired signal strength linearly. The measured bit error rate can be reduced 75 times with a long spreading factor. By taking advantage of this benefit, we propose some middle-ware solutions to monitor the network load and switch the spreading factors dynamically based on the current load with multimedia traffic. These middle-ware solutions are implemented in mobile and base stations, and experiments are performed to measure the actual system performance. The preliminary results indicated that our proposed system can always maintain a desired quality for all the voice connections.

CHAPTER 1
INTRODUCTION

Wireless networks make it possible for people to shop online anytime and anywhere. People will have instant access to information even when they are walking on the streets. In addition, customers will not only have access to plain information, but the information will be supported with multimedia components like images, animations, and audio. Recent technology advances are increasing multimedia capabilities in mobile devices. Cellular phones and notebooks are converging into a single mini-device which is capable of both computing and communicating. They are becoming more competitive to desktop PC in terms of computing power. In a near future, most of the multimedia applications will be able to run on mobile devices.

Therefore, a successful software/hardware system of M-commerce (mobile+e-commerce) needs to support a seamless integration to support voice, audio and other conventional data (e.g., e-mails, and ftp). The Table 1.1 lists the different traffic characteristics: We believe that ftp and e-mails require the highest support on the bit error rate (BER) since the accuracy of the content needs to guaranteed. However, there is no data rate requirement for these two traffic type (though short turn-around time is still preferred). Similarly image data (e.g., JPEG[1]) do not require a specific data rate, and can tolerate a higher degree of bit error.

---

[1] JPEG is a graphic file type used for displaying photographs on the Web. It stands for Joint Photographic Experts Group, after the group that created the standard.

Table 1.1: Bit error rate requirement of different traffic types

FTP	$BER < 10^{(-6)}$, no data rate requirement
$Email$	$BER < 7 * 10^{(-5)}$,no data rate requirement
$Image$	$BER < 2 * 10^{(-4)}$, no data rate requirement
$Audio$	$BER < 10^{(-3)}$, minimum data rate 10Kbps
$Voice$	$BER < 10^{(-2)}$, data rate 8Kbps–64Kbps

However, the bottom-line is that the minimum BER and data rate requirements for audio and voice types must be satisfied at any time. Otherwise the system will not guarantee the performance. Voice connections should be synchronized, and maintained at a certain range of data rate (typically 16-Kbps), any interruption will degrade the quality of voice. Audio traffic (e.g., streaming MP3) also requires a constant data rate (with even higher BER). We believe that M-commerce solutions need to support all of the above traffic types with guaranteed quality.

The ideal solutions of M-commerce should also support many users with guaranteed quality. These solutions should also transmit e-mails and ftp without connection re-establishment even in the middle of voice conversation. It is our belief that this application switching should be transparent. For example, during a conversation, if a user likes to send an email, he/she will be able to do that without disconnecting current voice application and re-establishing connection for its email application. To avoid 'connection' setup overhead, the user should stay connected no matter what application is running. We target our system and solutions to make these functionalities feasible.[2] In this thesis, we present our on-going study for the design issues to accomplish the above goals. We focus on the discussions in a single cell in this thesis, and our results will be extended to across multiple cells.

[2] The current cellular phones/PDA devices start offering e-mail and browsing functions. But these two functions can not be run with an active voice connection.

Today, the second-generation (2G) CDMA (Code Division Multiple Access) systems are installed worldwide. Advanced studies are continuing for the third-generation (3G) CDMA protocols. A next generation protocol is definitely needed to support higher data rates and better QoS. Directly putting different traffic to existing CDMA system has been shown to be a bad solution. Traditionally, voice-only-CDMA system has been designed based on the assumption of identical user channel capability. In this system, all the traffic channels are of the same data rate, and have the same bit error rate (BER). However this assumption may not result a good performance when multimedia communication is taken into account. Therefore, it is worthy to investigate schemes that can provide various QoS based on different traffic types. With such schemes, the overall system performance should be improved. Our work can be classified and developed into the new 3G W-CDMA (Wide-band CDMA) framework.

1.1 Motivation

One specific reason that the existing voice-only-CDMA systems is not flexible enough is because they are based on a predefined fixed spreading factor. The spreading factor of CDMA is the key variable in determining user data rate and associated BER. In theory, spreading factors make CDMA possible by repeating user's data signals such that they can be re-constructed at the receiving mobile stations. However, spreading factor is also responsible to produce interferences (called MAI[3]) between different users' signals. Spreading factors can be as short as 4, or can be as long as 160. The detailed discussions on MAI component are in chapter 3. In here, we only summarize the benefits for using longer spreading factors. Increasing spreading factor can benefit BER because it will increase the

[3] MAI stands for Multiple Access Interference.

desired signal strength linearly. The mean MAI caused by other users will decrease accordingly, and will approach to zero when the spreading factor approaches to infinity.

To have a clear picture of how spreading factors and the number of active users affect the BER, we performed experiments with this goal in mind. We adopted the Gold PN (Pseudo Number) sequence from IS-95. The Figure 1.1 depicts the BER under different number of active users and spreading factors. The experiments were performed within a single cell without interference from neighboring cells. Multi-path effects and thermal noise were not taken into account since we emphasize on the effect of spreading factor.

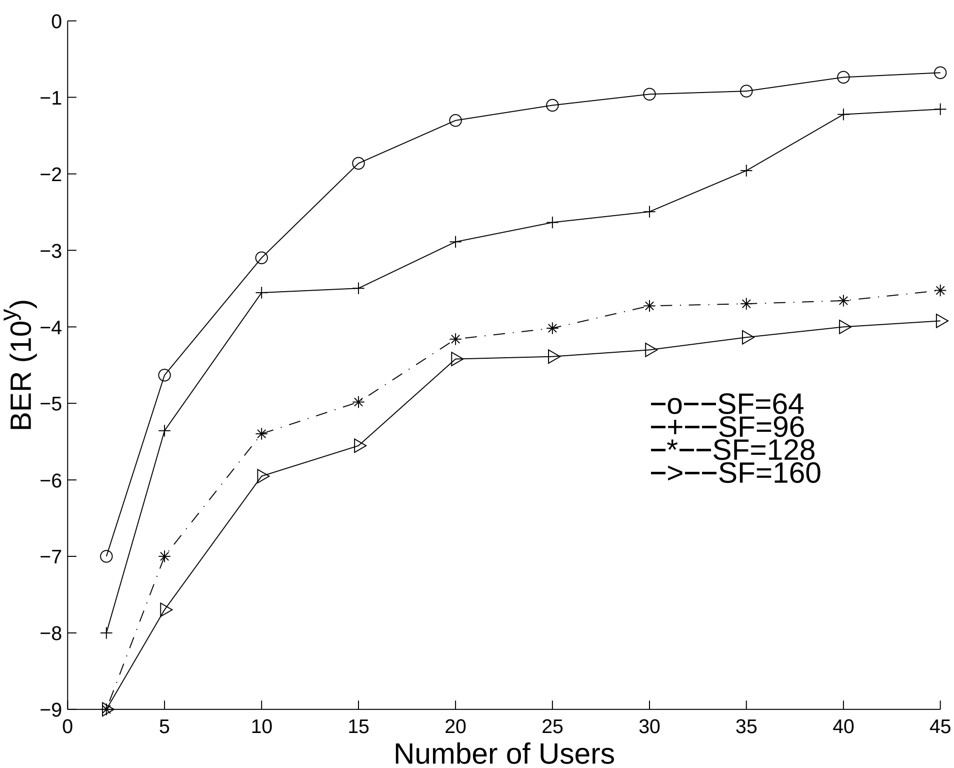

Figure 1.1: BER vs. User Number vs. SF

The performance results clearly indicate that the increase of spreading factor can effectively decrease the BER for a given number of users. For example, with 10 users, increasing the spreading factor from 64 to 96 will reduce BER from 0.0008 to 0.0003 (e.g., 62.5% reduction). Increasing spreading factor further to 128 results a BER of 0.000004, which is 75 times less. In order to support a variety of BERs, many hardware manufacturers are considering to bring in the dynamic capability of changing spreading factor in next-generation mobile and base stations.

Nevertheless, hardware capability only provides the possibility that different spreading factors can be jointly adopted. Hardware capability alone does not accomplish our goals as we identified. In order to achieve our goals, novel middle-ware solutions are required in mobile and base stations. The middle-ware solutions will monitor the network load and switch the spreading factors dynamically based on the current load from multimedia traffic. In this thesis, we particularly address the admission control protocol design based on the dynamic spreading factor.
All the protocol designs are based on IS-95 Channel Architecture for backward compatibility. Readers unfamiliar with the IS-95 Channel Architecture can refer to chapter 2 for detailed introduction.

Because of the possible adaptation of spreading factors, novel admission protocols (i.e., state diagrams) are proposed in mobile and base stations. With the new protocol, it is possible that a mobile user is notified to change the spreading factor. This usually happens when a new mobile presents an OPEN request. The preliminary results indicate that our proposed system always maintains a desired BER for all the connections (including the existing and newly-arriving one).

However, our protocol design requires additional procedures to be executed in the base station to make sure that the overall BER changes gracefully. Therefore a longer end-to-end delay for the newly-arriving connection may be experienced. We have analyzed the timing components, identified the performance bottlenecks, and proposed improved schemes to reduce the end-to-end connection setup time. By examining the CDMA characteristics, our schemes reduce the *admission processing time* component with a range of 22% and 27%. Using multiple Access Channels for acknowledging the successful change of spreading factors, the *update time* component was reduced with a range of 48% and 58%. Our results recommend that at least two Access Channels are required (current standard only requires one Access Channel) but no more than four with short spreading factors.

1.2 Related Study

A general survey of wireless MAC (Multiple Access Control) protocols is presented in a paper by Akyildiz et al. [2]. Protocols based on TDMA and CDMA are compared. Another paper discussed QoS guarantees in a wireless LAN with Dynamic Time-Division Duplexed (D-TDD) transmission [6]. Admission control and scheduling/multiplexing down-link packets are addressed. However, their work is based on TDMA mini-cell which is quite different from CDMA Cellular Network. In Nagaosa's paper, efficient code assignment and Multi-code Sense/CDMA is studied for Inter-Vehicle Communication Network (IVCN) to minimize total MAI [14]. However, this paper did not address the possibility of DSF (Dynamic Spreading Factor).

In a paper by Zhuge and Li, an admission control protocol for multi-service CDMA is developed based on interference estimation [17]. They used log-normal distribution to approximate the effect of random user location, shadowing, and imperfect power control. Pour proposed a MAC protocol to utilize the silent period of voice to transmit data traffic [15]. The data terminals compete for the available PN code from the relinquished voice terminal by slotted Aloha protocol.

In Choi and Shin's study, multi-code assignment is considered to provide various QoS for multi-type-traffic in cellular CDMA system [5]. Three types of traffic were defined and treated in a different manner, according to their unique traffic nature. Akyildiz also proposed a packet scheduling protocol for slotted CDMA [1]. The scheduler can maximize system throughput based on BER requirement. However, the system overhead of sub-dividing a CDMA frame-time is not considered.

In Oh and Wasserman's paper, system performance of a DS-CDMA when setting to a different spreading gain is studied [16]. Two kinds of traffic are considered. The author shows that optimal spreading gain is linearly increasing when the MAI increases. However, the work did not relate the control of dynamic spreading gain with the system load.

The evaluation of the BER has been studied intensively in several papers [7, 8, 13, 3, 9]. It is widely agreed that the BER is largely determined by MAI. In the paper by Fukumasa et al., the design of PN sequence is discussed to reduce MAI [8]. In another paper, the probability of successful packet transmission is analyzed for DS-CDMA system [9]. Choi and Cho proposed a power control scheme to minimize the interference of high data-rate users to the neighboring cells [4]. The result shows that the number of high data-rate users for data communication should be less than 6 in order to support enough voice user. However, these works did not cover the system BER with dynamic spreading factor. A new middle-ware protocol is needed to integrate different services more efficiently (i.e. voice, data, video).

In this thesis, we formalized BER behavior with different spreading factors and network load. As expected, the average BER increases proportional to the network load. It is also found that increasing spreading factor can efficiently increase the channel quality. This observation is used as a basis for our MAC protocol design. By monitoring the network load, our network dynamically adjusts spreading factor for different services.

The rest of the thesis is organized as follows: In chapter 2, we described the CDMA technology and current protocol standard in detail. In chapter 3, we formulated the BER behavior and related it to different spreading factors. In chapter 4, we described our proposed protocol in detail by demonstrating the state diagrams for both mobile and base stations. The performance measurements (BER behavior, admission delay) are also reported in this section. In chapter 5, we concluded our work.

CHAPTER 2
CODE DIVISION MULTIPLE ACCESS

Multiple access refers to the sharing of a common resource in order to allow simultaneous communications by multiple users, and this common resource is RF (Radio Frequency) spectrum. There are 3 basic categories of multiple access techniques.

In traditional FDMA (Frequency Division Multiple Access), each individual user is assigned a particular frequency band in which transmission can be carried out. A portion of the frequency spectrum is divided into different channels. Different user's signals are low-pass filtered and modulated onto an assigned carrier frequency f_c of a particular channel. This way, multiple users can simultaneously share the frequency spectrum. In TDMA (Time Division Multiple Access), each user is assigned a different time slot in which to transmit; in this case, the division of users occurs in the time domain.

In CDMA (Code Division Multiple Access), each user is assigned a unique code sequence it uses to encode its information-bearing signal. The receiver, knowing the code sequences of the user, decodes a received signal after reception and recovers the original data. This is possible since the cross-correlation between the code of the desired user and the codes of the other users are small. Since the bandwidth of the code signal is chosen to be much larger than the information-bearing signal, the encoding process spreads the spectrum of the signal and is therefore also known as spread-spectrum signal, and CDMA is often denoted as spread-spectrum multiple access.

In CDMA, each user's narrow-band signal is spread over a wider bandwidth. This wider bandwidth is greater than the minimum bandwidth required to transmit the information. The ratio of transmitted bandwidth to information bandwidth is called the *processing gain, G_p,* of the spread-spectrum system,

$$G_p = \frac{B_t}{B_i} \tag{2.1}$$

where B_i is the transmission bandwidth and B_i is the bandwidth of the information-bearing signal.

The spectral spreading of the transmitted signal gives to CDMA its multiple access capability. If multiple users transmit a spread-spectrum signal at the same time, the receiver will still be able to distinguish between the users provided each user has a unique code that has a sufficiently low cross-correlation with the other codes. All the spread wide-band signals (of different users) are added together to form a composite signal, and the composite signal is transmitted over the air in the same frequency band. Correlating the received signal with a code signal from a certain user will then only despread the signal of this user, while the other spread-spectrum signals will remain spread over a large bandwidth. Thus within the information bandwidth the power of the desired user will be larger than the interfering power provided there are not too many interferer, and the desired signal can be extracted.

2.1 IS-95 CDMA Channel Architecture

The cellular CDMA air interface is formally defined in IS-95 standard. IS-95 CDMA uses a bandwidth of 1.25 MHz and defines a set of channels [11, 12]. Communication occurs in forward direction (base station to mobile station) and in reverse direction (mobile station to base station).

The forward CDMA channel consists of a pilot channel, an optional synch channel, optional (to a maximum of seven) paging channels, and several forward traffic channels. Each of these channels is orthogonally spread by the appropriate orthogonal function (Walsh codes of length 64) at a rate of 1.2288 Mcps and is then spread by a quadrature pair of PN sequences at a rate of 1.2288 Mcps. All the channels are added together and sent to modulator. Walsh codes are used to distinguish different channels:

- **Pilot Channel:** Each base station transmits a continuous pilot channel. The pilot channel acts as a beacon to notify potential subscribers that a CDMA system is here. The pilot signal consists of the all zeros pattern and is modulo-2 added to the Walsh 0 function for the CDMA system It carries no information.
- **Synch Channel:** The synch channel is transmitted by a base station to enable the mobile station to obtain frame synchronization of the CDMA signal. The synch channel carries a repeating message that identifies the station, and the absolute phase of the pilot sequence. The message sent on the synch channel contains information about the base station and the serving CDMA system, such as the system identification (SID), the network identification (NID), the offset of the PN sequence for the base station and the long code state for that base station.
- **Paging Channel:** Base station uses this channel to notify mobile stations of incoming calls. It carries the responses to mobile station accesses. The paging channel sends many different types of messages. One of them is System Parameters Message, which is sent to all mobile stations in the area to specify the characteristics of the serving cellular system. Order Message directs the mobile station to perform an operation and confirms a request from the mobile station. Channel Assignment Message informs the mobile station of the correct traffic channel to use for voice or data.

- **Forward Traffic Channel:** After mobile station received a paging channel message and responded, a traffic channel is assigned dynamically to that mobile station. The mobile station is informed which code channel it is to receive via the paging channel message. Traffic channel carries voice or data traffic. IS-95 supports 9.6 or 14.4 Kbps data rates.

The reverse channel is the mobile-to-cell direction of communication. Different pseudo-random noise (PN) sequences are used to distinguish different channels. The signal is spread with a long PN sequence at a rate of 1.2288 Mcps. The reverse channel only consists of an access channel and several reverse traffic channels.

- **Access Channel:** The access channel is equivalent to paging in forward direction, but this time mobile station notifies base station that it wants to make a call. Page responses are also transmitted via access channel from mobile station to base station. Whenever a mobile station registers with the network, processes an order, sends a data burst, makes an origination, responds to a page, it uses that channel.
- **Reverse Traffic Channel:** Traffic channels are mobile-unique in the reverse CDMA channel. In other words, each station has a unique Long Code Mask, based on its electronic serial number. Whenever the mobile is assigned to traffic, it uses its specific long code mask. The frames transmitted are 20 *msec* long. IS-95 supports 9.6 or 14.4 Kbps data rates for data communication.

2.2 Call Processing

The IS-95 standard also specifies call processing protocols between base station and mobile station [10]. The state diagrams for terminal stations and base station are in Figure 2.1 and Figure 2.2 respectively.

Following basic steps are taken in mobile station when placing an outgoing call to base station.

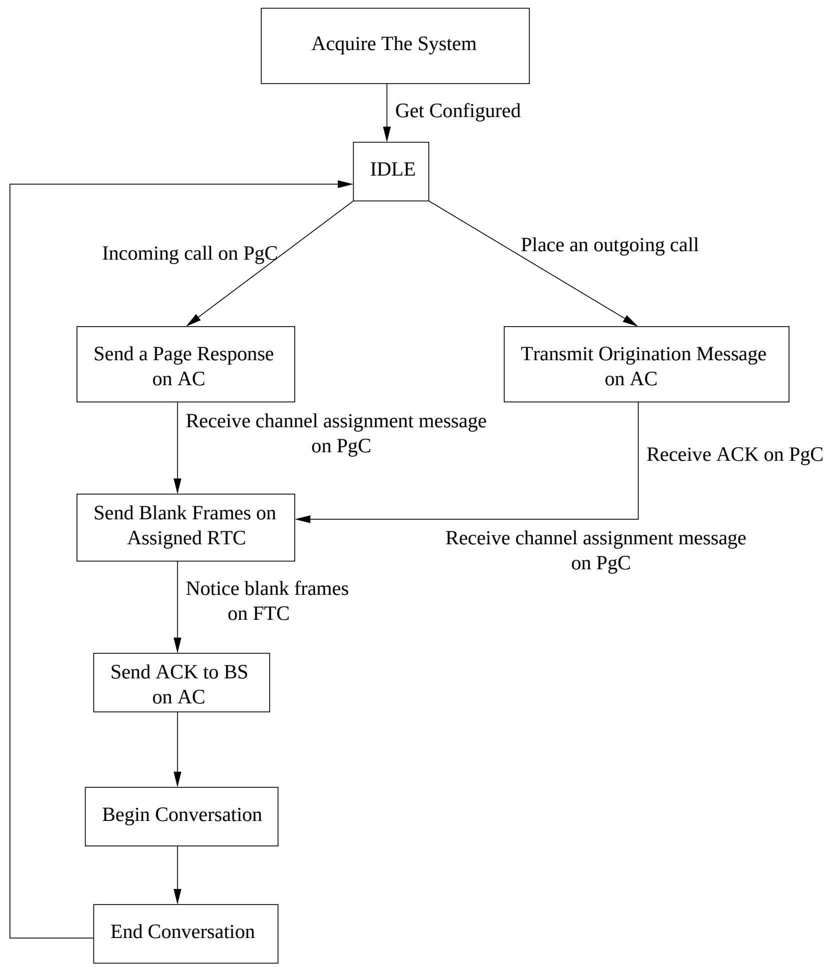

Figure 2.1: The state diagram for mobile station.

1. The mobile user dials the desired digits, and presses SEND.
2. Mobile transmits an Origination Message on the access channel.
3. The system acknowledges receiving the origination by sending a base station acknowledgment on the paging channel.
4. The system arranges the resources for the call and starts transmitting on the traffic channel.
5. The system notifies the mobile in a Channel Assignment Message on the paging channel.
6. The mobile arrives on the traffic channel.

7. The mobile and the base station notice each other's traffic channel signals and confirm their presence by exchanging acknowledgment messages.
8. The base station and the mobile negotiate what kind of call this will be.
9. The audio circuit is completed and the mobile caller hears ringing.

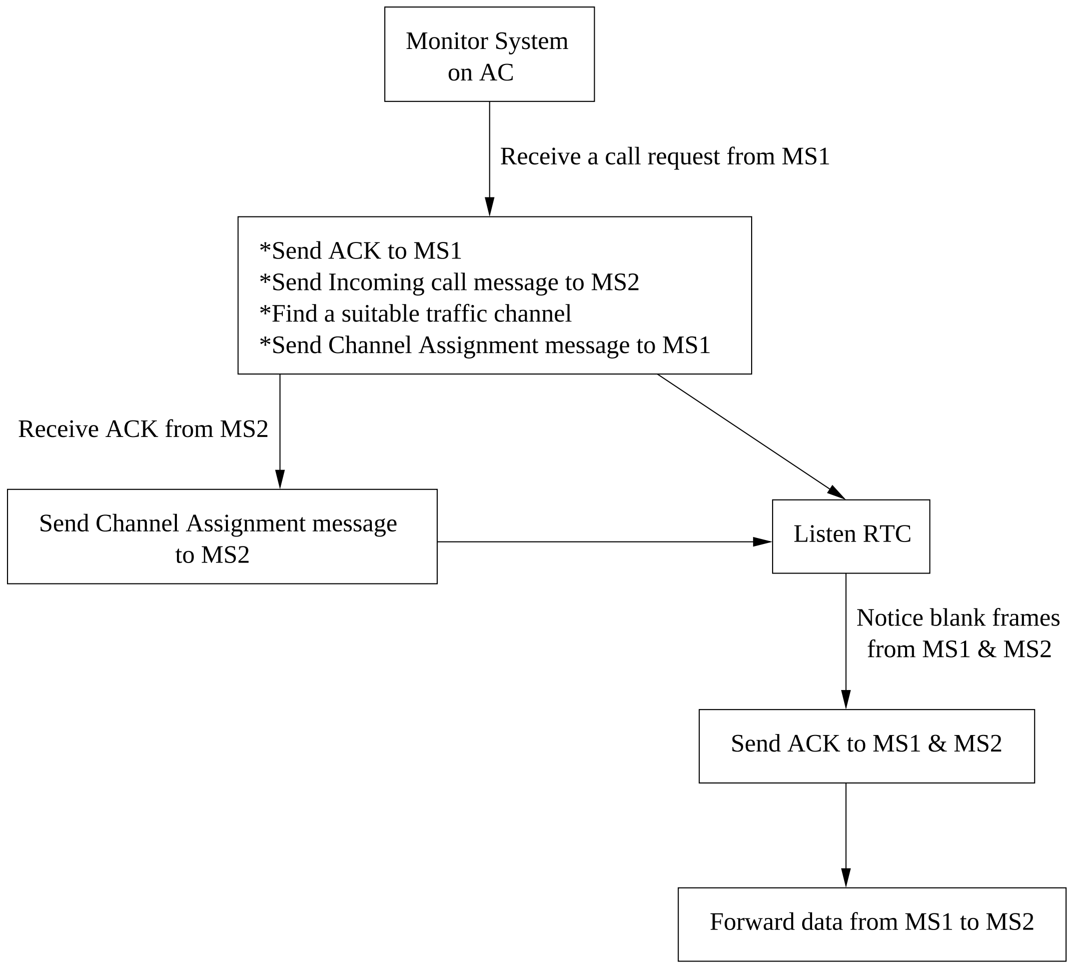

Figure 2.2: The state diagram for base station.

Following basic steps are taken in mobile station when receiving an incoming call from base station.

1. All idle mobiles monitor the paging channel to receive incoming calls.
2. When an incoming call appears, the paging channel notifies the mobile in a General Page Message.

3. A mobile which has been paged sends a Page Response Message on the access channel.
4. The system sets up a traffic channel for the call, then notifies the mobile to use it with a Channel Assignment Message.
5. The mobile and the base station notice each other's traffic channel signals and confirm their presence by exchanging acknowledgment messages.
6. The base station and the mobile station negotiate what kind of call this will be.
7. The mobile is told to ring and given a "calling line ID" to display

As it can be clearly seen, IS-95 does not consider any QoS issue when admitting new connections. The base station simple admits new connections as long as there is available traffic channel. However, the base station does not realize that increasing the number of connections increases the interference between users and therefore degrades the quality of user channels. The interference level is highly affected by the number of active channels. The proof is provided in the next chapter. However, we also proved that we can reduce the interference by using a larger spreading factor. The mathematical relation between spreading factor and MAI is also provided in the next chapter. Unfortunately, current IS-95 standard neither addresses the relation between the MAI and the number of active users nor provides a solution.

CHAPTER 3
ANALYSIS OF BER FOR DSF DS-CDMA

Previous work of BER analysis on CDMA system assume a fixed SF (spreading factor) for all users. When dynamic spreading factor assignment is taken into account, a new BER model for the uplink channel is desired. In the following discussion, we assume a single cell environment.

Let p^k be the spreading sequence (long PN sequence) of k^{th} user, the spreading signal can be represented as

$$p_k(t) = \sum_{n=-\infty}^{\infty} p^k(n)\delta_\tau(t - n\tau) \tag{3.1}$$

where $\delta(x)$ is the unit rectangle pulse with duration τ.

Let u^k be the data sequence of k^{th} user, the modulated sequence with a spreading factor of S_k is

$$u_k(t) = \sum_{n=-\infty}^{\infty} u^k(n)\delta_{T^k}(t - nT^k) \tag{3.2}$$

where $T^k = S_k * \tau$ is the duration of a data bit after modulation.

Then the k^{th} transmission signal will become

$$b_k(t) = p_k(t).u_k(t) \tag{3.3}$$

Assuming an additive white Gaussian noise (AWGN) with zero mean, perfect power control, and no multi-path fading environment, the received mixed signal should be

$$
\begin{aligned}
r(t) &= \sum_{i=1}^{K} b_i(t) + n(t) \qquad (3.4)\\
&= \sum_{i=1}^{K} \sqrt{P}.p_i(t).u_i(t) + n(t)
\end{aligned}
$$

For an interested user k, the correlate output is represented by

$$
\begin{aligned}
d_k(n) &= \int_{(n-1)T^k}^{nT^k} r(t)p_k(t)dt \qquad (3.5)\\
&= \int_{(n-1)T^k}^{nT^k} r(t)p_k(t)dt\\
&= \int_{(n-1)S_k\tau}^{nS_k\tau} \{\sum_{i=1}^{K} b_i(t) + n(t)\}p_k(t)dt\\
&= \frac{\sqrt{P}.S_k.u_k(n)}{2} + \sqrt{P}\int_{(n-1)S_k\tau}^{nS_k\tau} \sum_{i\neq k} u_i(n).p_i(t).p_k(t)dt +\\
& \int_{(n-1)S_k\tau}^{nS_k\tau} n(t)p_k(t)dt
\end{aligned}
$$

Here P is the received power of transmitter's signal and $n(t)$ denote noise. A maximum likelihood decision rule based on $d_k(n)$ is used at receiver, thus the decoded n^{th} bit of user k is

$$
u_k\hat{(n)} = \begin{cases} 1 & \text{if } d_k(n) > 0 \\ -1 & \text{else} \end{cases} \qquad (3.6)
$$

The bit error probability is thus $Prob_k(n) = Prob(u_k\hat{(n)} \neq u_k(n))$. From equation (3.5) and (3.6), it is clear that $Prob_k(n)$ is determined by the strength of desired signal (the first component), the MAI (second component), and channel noise (third component) of equation (3.5). When the signal is strong enough (i.e, signal/noise ratio is large), it is well known that MAI is the dominant factor of decoding error. Therefore we will focus on the signal strength and MAI components in the following discussion.

The desired signal $\frac{\sqrt{P}.S_k.u_k(n)}{2}$ is the linear amplification of the original signal by the factor of spread factor S_k. It is intuitively true that a larger S_k will result a stronger desired signal, therefore reduce the BER for that particular user.

The MAI for the user k will be

$$\mathcal{I}_k = \sqrt{P} \int_{(n-1)S_k\tau}^{nS_k\tau} \sum_{i \neq k} u_i(n).p_i(t).p_k(t)dt \tag{3.7}$$

Numerous research have proved MAI ($\mathcal{I}_k$) as the major interference to the user signal. Equation (3.7) shows that $\mathcal{I}_k$ is determined by the cross-correlation of $p_k(t)$ and other spreading sequences $p_i(t)$. When $p_k(n)$'s are i.i.d (independent identical distribution) and have a mean value $\mu = 0$, the expectation of $\mathcal{I}_k$ is also 0. In fact $p_k(t)$'s are all PN sequences, which are not i.i.d. Nevertheless, many pseudo random (PN) sequences have small cross-correlation, such as gold-sequence (used in IS-95). Due to the nature of pseudo random sequences, the longer the length of convolution, more likely two PN sequences will be independent. Therefore we can expect that $\mathcal{I}_k$ will approach to zero when S_k increase.

Figure 3.1 depicts the cross-correlation between PN sequences under different spreading factor. Since the cross-correlation is applied over part of the PN sequences, we average the measurement for several random tests. For SF=32, the mean $\|\mathcal{I}_k\|$ is about 0.507. When SF increase to 64, the mean value of $\mathcal{I}_k$ reduce to 0.283, about 44% less than that of SF=64. Further increasing in SF (96 and 126) result mean cross-correlation of 0.167 and 0.047 respectively, which is 3 and 11 times less than the case of SF=32. The reduction in cross-correlation of the received signal increases the overall signal/noise ratio, thus a better BER can be expected.

We have shown that a larger spreading factor will improve the channel quality. The problem transforms to designing a new multiple access protocol. The protocol should make the base station more intelligent so that it will take necessary steps

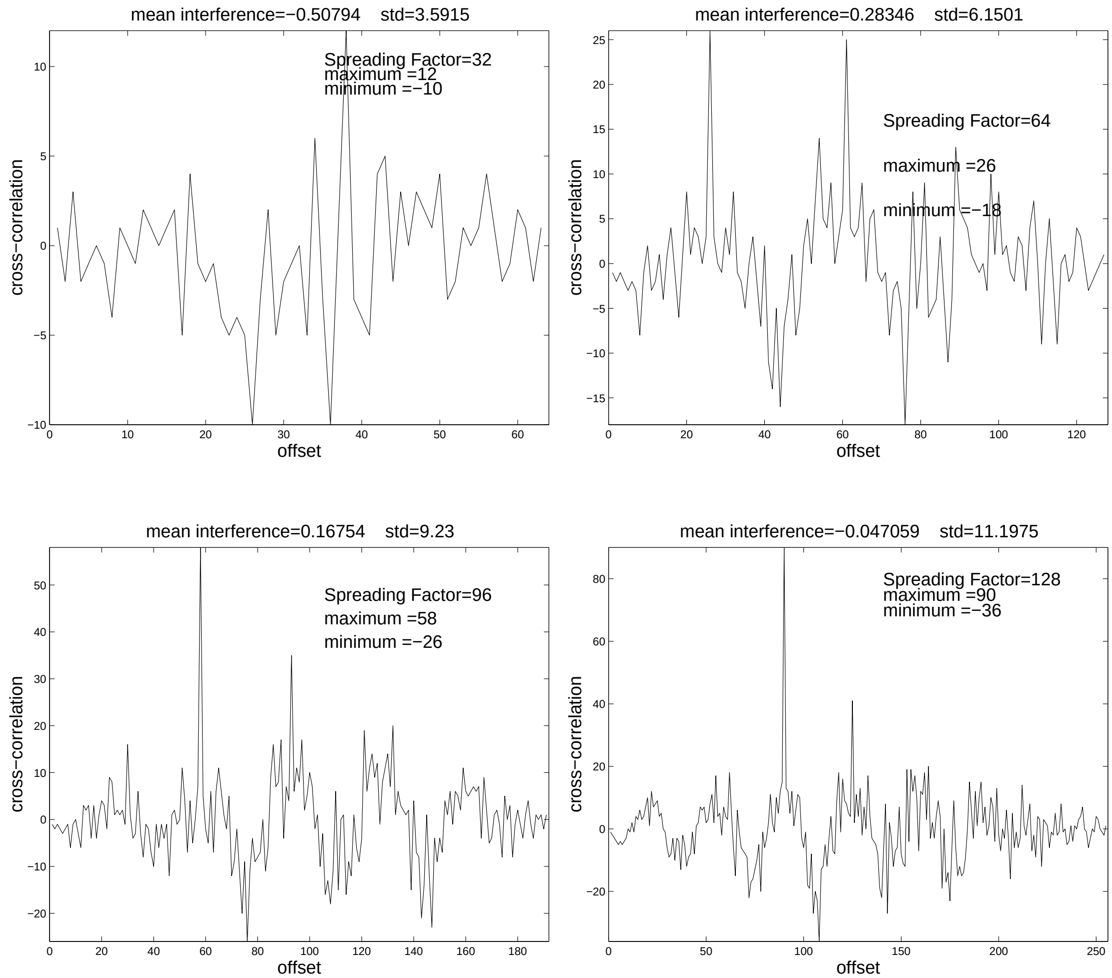

Figure 3.1: Distribution of MAI (N=2).

to maintain the quality of open channels when the number of connections increase. We provided a solution at the admission control level. The base station decides (the decision is made during the admission of a new call) if it is necessary to take an action and order mobile units to use the appropriate spreading factor in order to maintain quality. In the next chapter, we will discuss our admission control protocol in detail and do performance analysis.

CHAPTER 4
PERFORMANCE-GUARANTEED MAC PROTOCOL

The ultimate goal of admission control protocol is to support as many users as possible while still satisfying BER requirements for all existing connections. Conventional FDMA schemes divide the frequency spectrum into multiple channels. Every user's connection needs to allocate one channel from the base station before the voice conversation can take place. When all the channels are allocated, no more connections can be admitted. Therefore, admission control with FDMA schemes is straight forward.

However, CDMA-based schemes (along with the multimedia support and dynamic spreading factors) make the admission decision non-trivial. One of the advantages of CDMA system is that it does not need the particular channel allocation. Every connection can be established by their own PN code. Thus determining an exact point to block newly-arriving connections are difficult. In addition, our system allows the users to change the traffic types without re-establishing the connections. Therefore, once a call is accepted, the performance will be guaranteed during the life time independent of the other existing or newly-arrived connections.

It is clear that there are many factors that can influence the performance of the call depending on the duration of the call. It is even possible that during a long call, the system will accept and terminate a number of calls, thus the number of users will alter very often. Those alterations will affect BER at any given time. Thus our admission control protocol needs to be adaptive for these changes. Our proposed admission control protocol is able to monitor and assure that performance remains almost the same by taking necessary steps whenever needed. As our system admits new calls (thus increasing the number of active users), system will dynamically change the spreading factors, if necessary, for certain traffic.

Both mobile and base stations need to participate in the call admission process. Mobile stations are the ones that make the requests and/or update their parameters to adjust themselves to changing environments. The environment change includes increase/decrease in the number of users and alterations in traffic types. As an example, a station may start a connection with voice, halt the voice without terminating the connection, send email, and go back to voice communication. This scheme eliminates the significant overhead (in term of tens of seconds) caused by terminating and re-opening a new connection. Therefore, the admission control protocol that we propose here is flexible and capable of guaranteed overall system performance.

4.1 Proposed Admission Protocol in a Mobile Station

The Figure 4.1 depicts the state diagram (i.e., protocol) for the mobile stations.

Unlike traditional CDMA systems (IS-95), mobile stations can have 3 types of request: OPEN a new connection, ALTER the traffic type, and CLOSE the connection. The first two request types follow the similar steps except the fact that altering the traffic type does not change the number of users. The application in mobile unit sends an OPEN request to base station on the Access Channel. Along with the request the type of the traffic (and the desired minimum data rate, if required) is sent. Our protocol defines 4 traffic types (VOICE, AUDIO, VIDEO, DATA) and multiple data rate options for each traffic type. IS-95 does not allow mobile to specify the desired traffic type.

The OPEN procedure is covered with a dashed line. The mobile chooses one traffic type and one of the data rate options provided. Once the base station

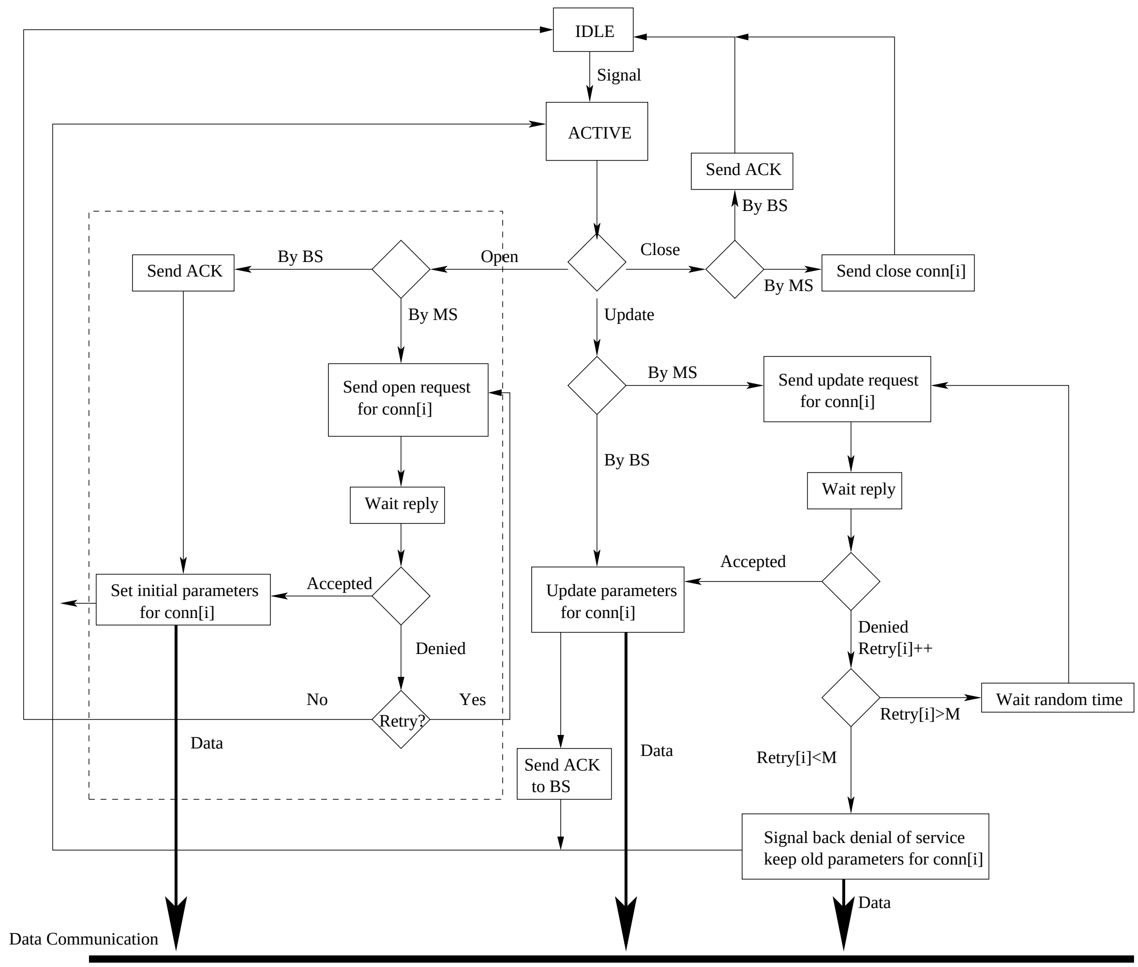

Figure 4.1: The state diagram and protocol of a mobile station.

receives that request, it first checks if the new request can be satisfied independent of other connections. The satisfaction is determined based on two things: The type of the traffic to be carried on the new connection, i.e. the BER requirement and minimum data rate, and interference by other users. The minimum spreading factor that will provide the BER satisfaction is found for the new connection, and maximum data rate allowed by that spreading factor is calculated. This is compared with the minimum data rate required by the specific application and the first decision is made. The decision is either to continue remaining steps or to deny the request. If the request is denied, the mobile is allowed to retry after waiting a random time period.

4.2 Proposed Admission Protocol in a Base Station

The Figure 4.2 depicts the state diagram (i.e. protocol) for the base stations. Our major contribution to the new protocol design is covered with a dashed line. If it is determined that the new connection can be satisfied, the base station moves further to check if existing connections can be satisfied once the new connection is up. This facility is not available in IS-95 CDMA. However, it is added to our protocol to ensure quality for all users. For each existing connection, the system calculates the expected average BER corresponding to the increased number of users. If the expected average BER is high, we try to increase the spreading factor and check the maximum data rate in order to see if data rate requirement can also be satisfied with an increased spreading factor. This step is repeated till all existing connections are reviewed.

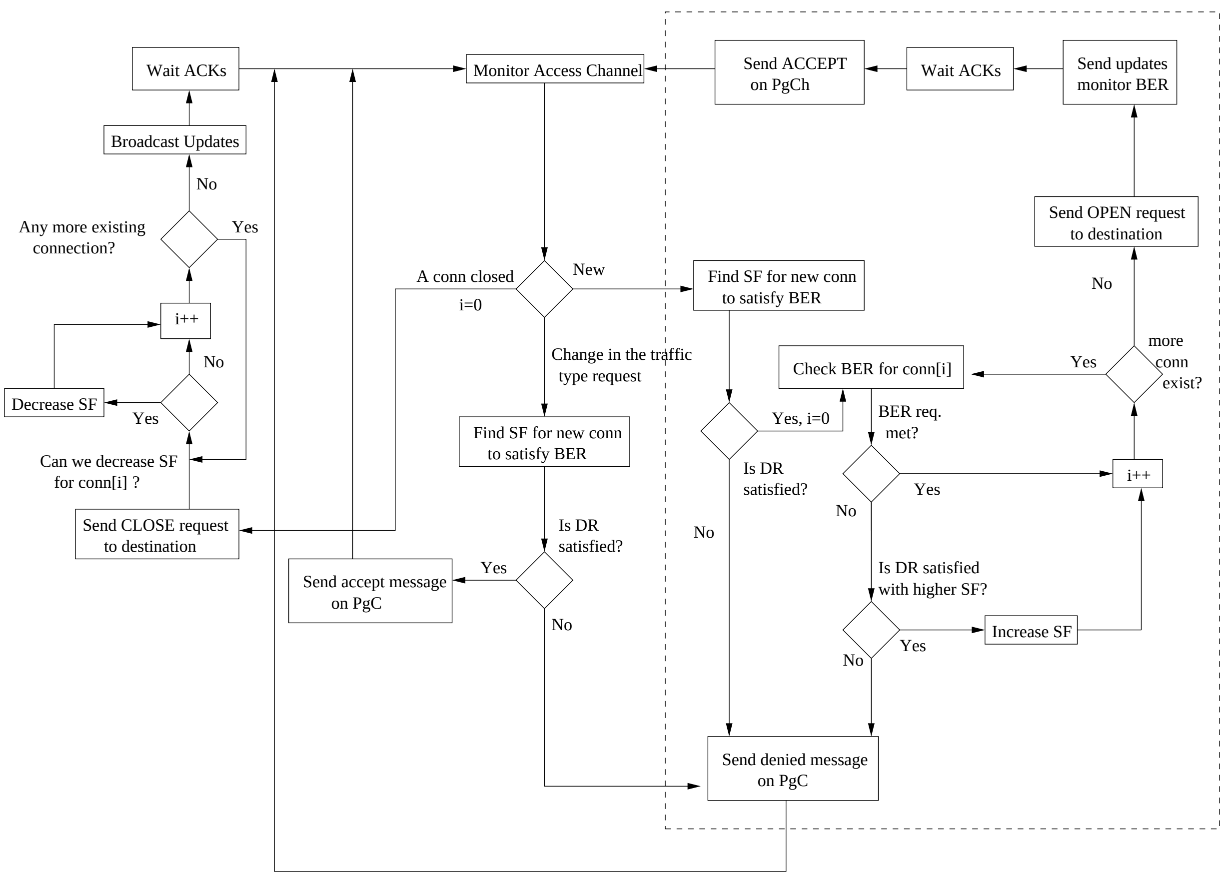

Figure 4.2: The state diagram and protocol of a base station.

We can have two situations after the review: First, no existing connection requires an update. The destination mobile is immediately informed of OPEN request and an ACK is sent back to the requesting mobile. The second situation is that some connections may require an update. For each traffic type requiring an update, the base station broadcasts an UPDATE message. All mobiles using the same traffic must send an ACK back to base station to confirm that they have updated their parameters. This is necessary to make sure that no one suffers bad performance once the new connection is active. The update procedure is a major improvement on IS-95. It is the key in providing the quality to all calls. This procedure allows the mobiles to adjust their spreading factors dynamically. When all ACKs are received, the base station sends an ACK to requesting mobile, and the mobile may start to transmit its data.

The mobile stations send a CLOSE request to terminate a connection. Contrary to OPEN request, CLOSE requests cause a decrease in the system load. Base station decrements the total number of users and review each existing connection. The purpose of the review is to attempt to shrink spreading factors. Since the traffic load is reduced as a result of connection termination, the average BER will drop and shorter spreading factors may satisfy the BER requirements for certain connections. Decreasing spreading factor for a connection will provide more bandwidth for that user. During review, base station will do same operations as it did for an OPEN request, but this time it will consider a decrease in the number of users while doing its computations. If it is possible for any traffic type to use a smaller spreading factor, base station broadcasts an UPDATE message. Mobiles communicating with that traffic type must send an ACK back to base station to confirm that they have updated their parameters.

In this thesis we have investigated the expanding phase only (covered with dashed lines in both state diagrams 4.1 and 4.2). We haven't done any deep analysis on the shrinking phase (CLOSE requests). We hope that the same procedures in the expanding phase will also apply to the shrinking phase. We assume that the result of performance analysis on the shrinking phase will be similar to the ones that we had on the expanding phase. We leave the testing correctivity of our assumption as a future study.

Another important functionality of the protocol is its flexibility to alter the traffic type. You cannot switch from one traffic to another in IS-95 without terminating and re-establishing the connection. This function is provided only if a connection is already established. If the mobile decides to change the traffic type carried on traffic channel, it will send an UPDATE message to the base station. The message has to specify the requested new traffic type. The base station will again follow the same steps as it did for an OPEN request, but this time it will not consider an increase in the number of users while doing its computations.

4.3 A Performance-Guaranteed System

The added and/or improved functionalities bring new challenges to the implementation of the protocol as well as better performance for data communication. We have implemented the prototype of the proposed protocol. We performed an emulation of voice traffic up to 50 users for measuring the performance of our admission control protocol. We have only considered an expansion in the system load (i.e. subsequent OPEN requests). Practical parameters associated with the hardware and software environment in the mobile and base stations are listed in the following Table 4.1.

The spreading factors used in the experiments are 32, 64, 96, 128, and 160. The experiments presented in this section only tested homogeneous voice traffic. All connections use the same data rate. The BER requirement for voice traffic is

Table 4.1: Practical parameters used in the wireless CDMA environment

Chip Rate	4.096 Mcps
Packet Size	128 bits
Spreading Factors	32,64,96,128,160
Paging Channel Data Rate	32 kbps
Access Channel Data Rate	16 kbps
BER for voice	10^{-2}
Min Data Rate for voice	8 kbps

assumed to be 10^{-2}. The minimum data rate required by voice is 8-Kbps[1] . For simplicity, packet size is fixed to 128 bits.

Since the spreading factor reduces the information bandwidth and IS-95 bandwidth is not enough for multimedia traffic, we used the chip rate proposed in the emerging W-CDMA (Wideband CDMA) standard. W-CDMA proposes a chip rate of 4.096 Mcps. The maximum traffic channel data rate in this case will be 128 kbps with a spreading factor of 32. The data rates of Paging and Access channels are 32 Kbps and 16 Kbps respectively. Same spreading factors are used for Paging and Access channels as defined in IS-95. BER was measured in every mobile station, and the average BER among all the voice streams was calculated and illustrated in the following Figure 4.3.

As depicted by the curve labeled as "SF32", fixed-spreading-factor schemes did not guarantee the overall performance among all the users. When the connections are less than five, the average BER was acceptable (i.e., less than 10^{-2}). However, when the connection number exceeded five, every connection (including existing and the newly-accepted connections) suffered with the BER quality above 10^{-2}.

On the other hand, by using our proposed admission control protocol, the average BER was always maintained below 10^{-2} threshold for voice even as the

[1] Our latest study (along with Choi and Shin [5]) indicated that this is possible with advanced compression schemes such like GSM.

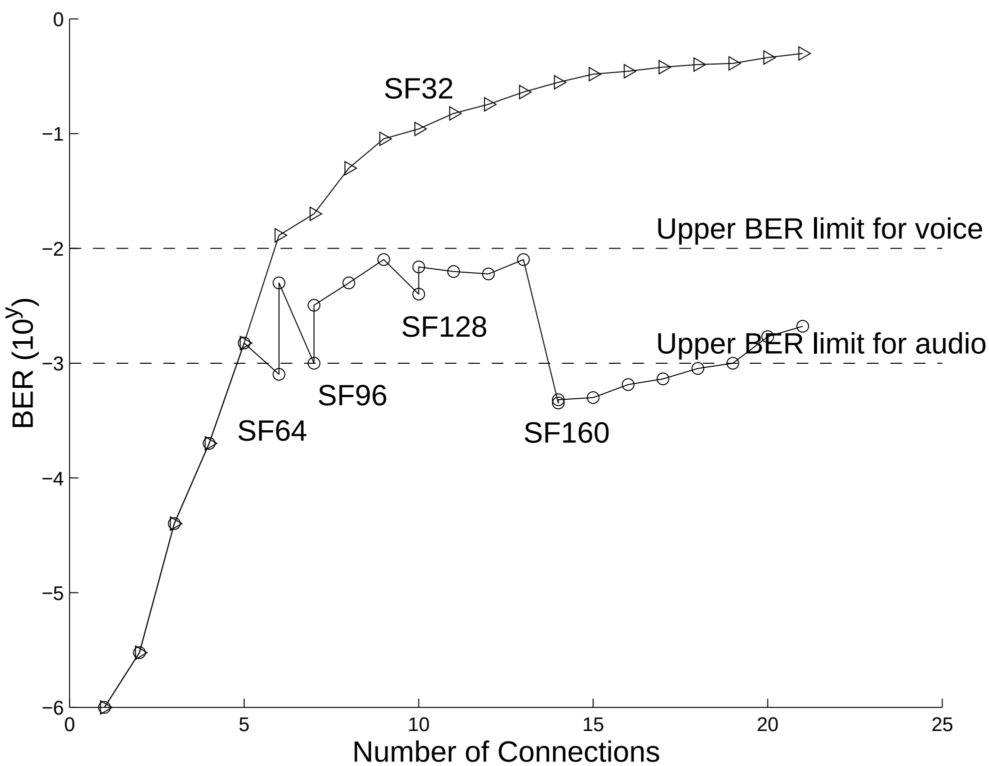

Figure 4.3: Performance guarantee with the proposed admission control.

number of user increased. The curve labeled with a sequence of (SF64, SF96, SF128, SF160) in Figure 4.3 stated the time instances that our CDMA system re-acted to the increasing demand of users, and new spreading factors have been adopted. The system never exceeded the upper limit for the voice traffic, (10^{-2}). The following items listed the detailed analysis:

- Starting with no connection the first connection is established between two users and spreading factor of 32 is used. The average BER is 10^{-6}. A new connection increases the average BER to $3 * 10^{-6}$ (i.e., the quality was degraded, but still acceptable). As new connections are established, the average BER keeps increasing dramatically.
- When the system had 5 active connections, the average BER is $1.5 * 10^{-3}$, 1500 times higher (i.e., worse in terms of quality) compared to the single connection at the beginning.

- When the number of active connections was maintained at 5 and a new connection request arrived, our proposed admission protocol sensed that the average BER would be degraded to violate the 10^{-2} satisfactory point for voice with 6 connections. Therefore the existing 5 connections should adjust their spreading factor to 64.
- The base station thus informed all the existing 5 connections (i.e., 10 mobile stations) to change their spreading factors. While the base station waits for acknowledgments from all mobile stations, the average BER will drop slightly since users start using a mixture of spreading factors. Once all acknowledgments are received, the average BER will drop to $(7.9 * 10^{-4})$, which is the bottom for spreading factor 64 and 5 existing connections. The drop in average BER causes the new connection to be accepted.
- As a result, the average BER increases again since the number of users are increased once the new connection becomes active. However, the average BER with spreading factor of 64 for 6 connections was measured as $(5 * 10^{-3})$, which is sufficient for voice communication.
- At this time another connection request arrives, and the base station determines that SF 64 will not be enough to support any more users. The steps described in previous situation are repeated and the average BER is again maintained below the voice limit.
- Further spreading factor changes are needed when number of connections become 10 and 14. Each time 2 steps are required to complete admission. First base station determines the new spreading factor, notifies all users, waits acknowledgments and finally activates the new connection.
- Note that the average BER usually drops to bottom while users adjust their spreading factor, and increases again once the new connection becomes active. The improvement in average BER becomes more significant as the spreading

factor increases. For instance, using SF 160 can improve the average BER at least from 10^{-2} to 10^{-3}, which achieved better quality. In addition, as the spreading factor increases, the rate of increase in BER (because of new connections) becomes less and less.

Though the experimental results are very promising, our proposed admission control protocol does introduce few consequences in two design tradeoffs. One of the tradeoffs occurs in the longer processing time at the base station. The other tradeoff occurs in the contention time for all mobile station to acknowledge the accomplishments of changing spreading factors. These two timing factors thus result a longer end-to-end connection setup time. During the next sections, we describe these trade-offs in details while proposing methods for further improvement.

4.4 Design Tradeoffs

The end-to-end delay is defined as the total time between a user making a connection request and the user becoming ready to transmit data. We can formulate this as follows:

$$T_{endtoend} = T_m + T_p + T_d + T_{update} + T_a + T_s, where \tag{4.1}$$

T_m = Time to send request from mobile to base station using Access Channel.

T_p = Base station processing time, includes finding spreading factor and checking existing connections.

T_d = Notifying destination, and receiving ACK from destination.

T_{update} = Broadcasting UPDATE message, and receiving ACKs.

T_a = Sending ACK to requesting mobile using Paging Channel.

T_s = Mobile updates channel parameters and becomes ready to transmit.

On the first design trade-off, the base station has a high computation complexity (i.e., T_p). While the performance of all connections are guaranteed, the base station's processing time increases proportionally to the traffic load of the system. The reason is that the base station checks every existing connection one by one. It also has to do some computation, like finding appropriate spreading factor, calculating the data rate, and doing comparisons. It has to repeat those operations for all connections if an update is necessary.

4.4.1 Processing Time at Base Station

We used our prototype system to measure the complexity on a base station. The numerical results are shown in the following Figure 4.4. A new connection arrives every time for different number of existing connections, in other words, under different load.

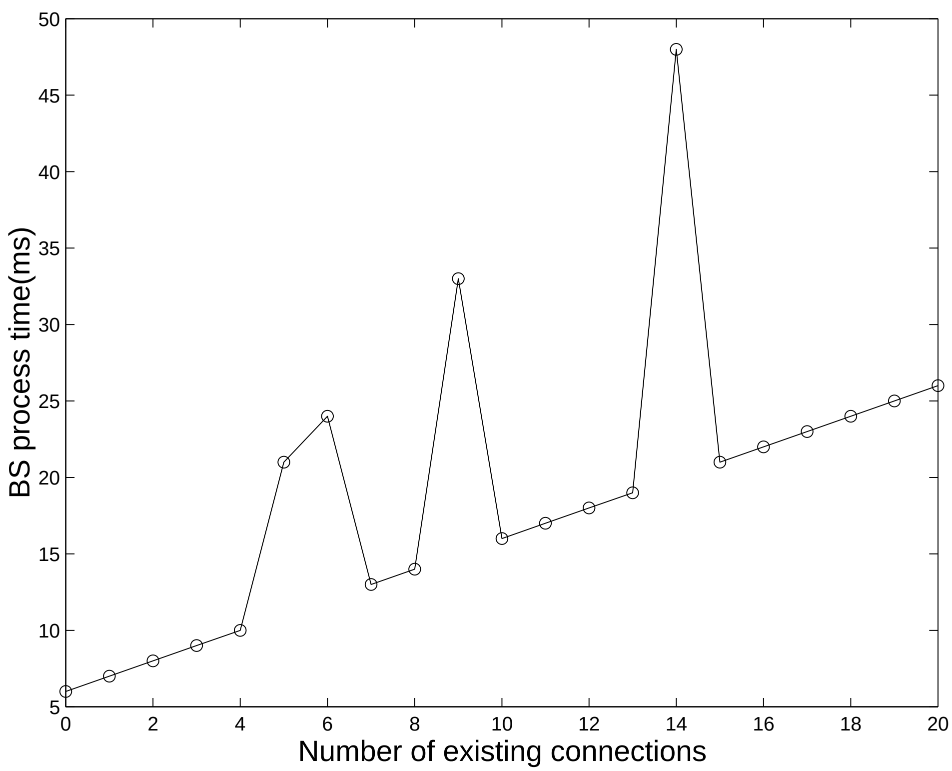

Figure 4.4: Admission processing time in a base station.

In general, each change of spreading factor will result a higher computational time than the previous one because more existing connections need be to examined and informed. Fortunately, the process time in the base station increases with a slight angle for most of the time. The reason is that, if the current spreading factor for existing connections is large enough, there is no need to decide new spreading factors for those connections. However, there are high peaks occasionally. Those peaks occur when accepting new connection means bad performance for existing ones. In this case, the base station attempts to find a suitable spreading factor, such that the performance is guaranteed. The following paragraphs describe the detailed interactions by accepting more voice connections.

Starting with an idle system (no active connections), the base station spends 6 milliseconds to process the first connection request. The majority of the time is spent to search for a proper spreading factor for the new connection and to do comparisons for data rate satisfaction. When the second request arrives, the system already has one active connection. The CPU time increases only by 1 *msec* because the base station has to review one existing connection in addition to find a spreading factor and do comparisons for new connection. Hence the whole process time is 7 *msec*.

The CPU time to process a new connection request will increase in average by 1 millisecond as the system load increases one by one. The admission protocol takes approximately 1 *msec* to review one connection if it is determined that the connection can be satisfied. However, when the need for a new spreading factor is required, the required time is much higher. For instance, when the number of existing connections reaches 5, the process time increases significantly to 22 *msec*.

According to our admission protocol, a change of spreading factor is required for existing connections. The base station has to find a new spreading factor and do data rate check for each connection. Now it takes 3 milliseconds to review each connection since each requires additional computation due to the needed adaptation. The processing time will further increase when the number of existing connections becomes 6, 9, and 14 (i.e., 24 *msec*, 34 *msec* and 48 *msec*).

4.4.2 Improved Schemes for Reducing T_p

Further improvement to reduce the admission processing time is possible (at least for the homogeneous voice streams). As we investigated the characteristic of situations, we realized that the connections belonging to the same traffic type usually change to the same spreading factor. This observation shows that the average BER is not affected by the type of the traffic. All external signals from other types of traffic are treated as noise no matter what type of traffic they carry. Thus, connections with the same BER requirement (same traffic type) are recommended to use same spreading factor.

Originally, our protocol was calculating spreading factor connection by connection in case of an update. There were two operations performed during each review, finding spreading factor and checking data rate satisfaction. The protocol can be further improved by calculating only one spreading factor for each traffic type before starting the whole review. This improvement saves an extra operation (without affecting the protocol correctness) during the review of each connection. The improved results are shown in Figure 4.5.

For the case of 5 connections, the processing time has been reduced from 22 *msec* to 17 *msec* (i.e., 22.7% reduction on T_p). Similarly, the processing time has become 18 *msec* for the case of 6 connections (i.e., 25% reduction on T_p). The improved degree can be more significant for the cases of 9 and 14 connections. The reduction reaches to 26.5% (i.e., from 34 *msec* to 25 *msec*) for 9 connections, while 27.1% is achieved (i.e., from 48 *msec* to 35 *msec*) for the case of 14 connections.

We consider the improved T_p is close to the optimum that a system can achieve. It is necessary that our system reviews each connection since each of them may have different data rate requirement. Nevertheless, we have reduced the T_p time during the review to a minimum, which is now only checking the data rate satisfaction.

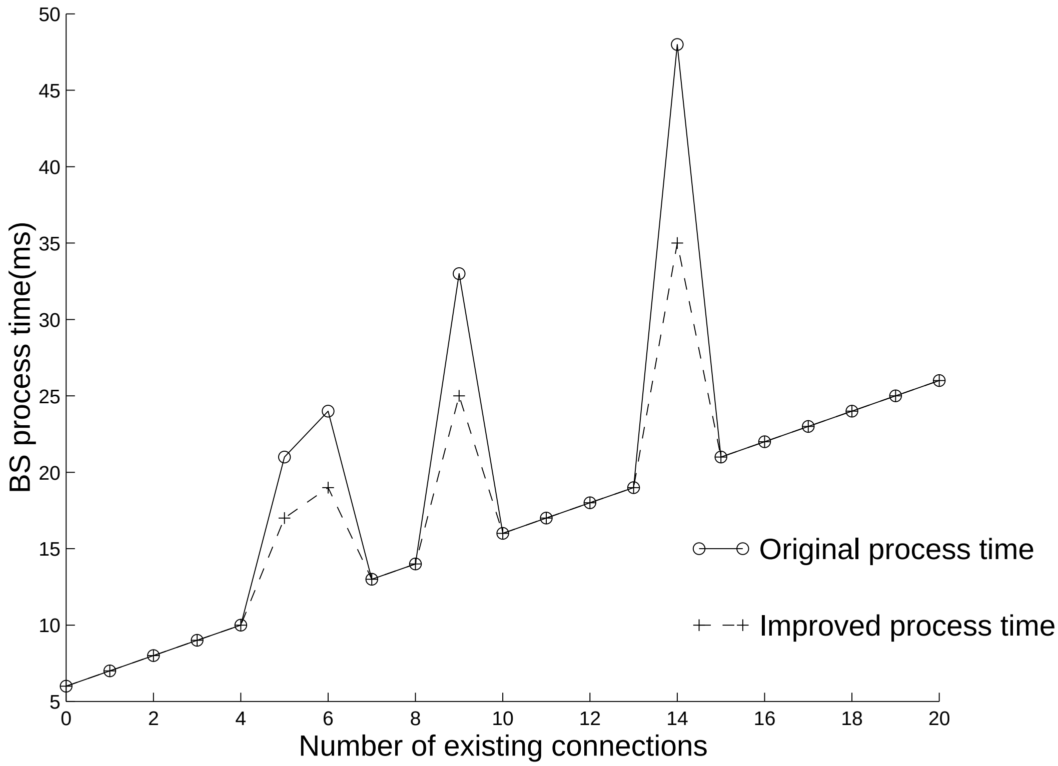

Figure 4.5: Improved process time(T_p).

4.4.3 Contention Time for Acknowledgment

The second tradeoff is the T_{update} component. After we trace the detailed timing analysis, we have observed that the dominating factor of T_{update} is because of the possible UPDATE broadcasts. This tradeoff is affected when existing connections require a change of spreading factor (thus an update for the overall system parameter). Our goal of the system design is to provide good BER performance for all existing connections at any time. Therefore, before accepting the new connection, all active mobiles should switch to the new spreading factor so that BER never increases above the acceptable point. The current system design enforces that mobiles should send acknowledgments back to base station after they change their spreading factors. Since the Access Channel of a CDMA system has only one common channel, this common channel is shared by all the mobile stations.

Thus, there will be extra delays due to possible collisions, if more than one mobile stations need to access the common channel.

The current protocol design in the base station will wait until it receives all acknowledgments from the mobile stations. Therefore, the duration time that after all collisions are resolved plays a very important role in the $T_{endtoend}$. The numerical results are shown in the following Figure 4.6. The graph is similar to the one in Figure 4.4. However, the peaks are much higher due to the dominating component T_{update}.

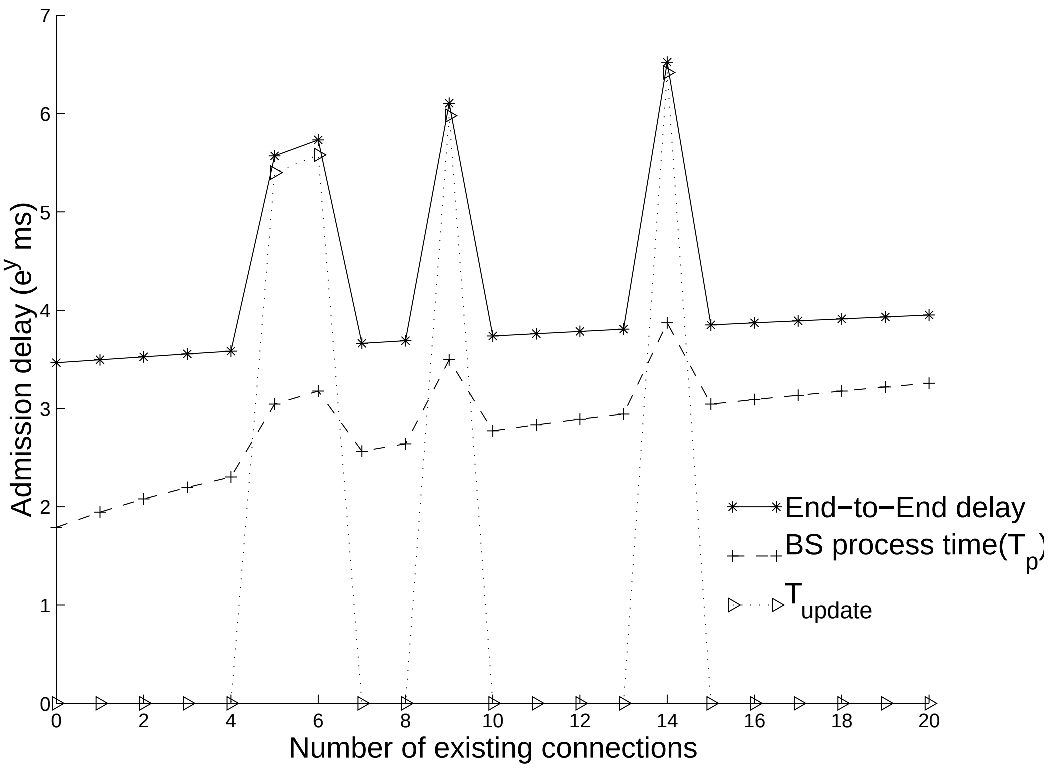

Figure 4.6: End to end admission delay.

In general, $T_{endtoend}$ seems to be determined by the combination of two dominating time components, T_p and T_{update}. T_p always exists and has been analyzed in the previous subsection. On the other hand, T_{update} does not occur all the time. For instance, T_{update} is 0 when no update is needed. Since the base station does not broadcast UPDATE message, the mobile stations thus do not required to send back any acknowledgments. Therefore, the total end-to-end delay grows slowly when no updates are needed.

However at every update point end-to-end delay and T_{update} grows exponentially for a short period, and then settles back to small delays. The first update occurred when number of the existing connections is 5. T_{update} becomes very large compared to the base station's processing time. The base station sends a broadcast message and all mobiles try to transmit acknowledgments almost at the same time using the common Access Channel. This will cause large number of collisions depending on the number of users competing for Access Channel. Therefore a collision resolution protocol is necessary when multiple users are trying to send ACKs to the base station.

In our experiments, we illustrated this using Binary Exponential Back-off algorithm. Analytically, average number of slots per contention is $\frac{1}{A}$ where A is the probability that some station acquires the channel in a specific slot. The maximum value of A is $\frac{1}{e}$, and average number of slots per contention becomes e. The total delay grows to 262 *msec*, and it's T_{update} component becomes 221 *msec*. When number of existing connections is 9, the end-to-end delay becomes 448 *msec*, and T_{update} is 395 *msec*, but when number of existing connections is 10, the end-to-end delay goes back to 42 *msec* and T_{update} is 0.

4.4.4 Improved Schemes for Reducing T_{update}

Certainly this half-second delay is not a desired feature. Therefore, some methods need to be sought for possibly reducing the T_{update} time component. We have analyzed the overall system behavior, and discovered that the UPDATE delays can be reduced by decreasing the contention period. Many approaches can potentially decrease the contention period. For examples, the goal can be accomplished if an improved collision prevention/resolution algorithm is used (i.e., other than CSMA[2] methods). Increasing the number of Access Channels is also another alternative. Without any technology preference, we have first investigated the second approach by increasing the number of Access Channels. We will explore the other approaches as research continues.

In the improved system protocol, we increase the number of available Access Channels for the mobile stations. Since now more-than-one Access Channels are available, all mobile stations need to be assigned to a specific Access Channel when they are admitted by the base station. A base station should distribute Access Channels among all users equally, thus the overall T_{update} can be reduced evenly across all the mobile stations. A hash function implemented in the base station should be sufficient for this purpose. By taking a mobile station's serial number or PN, we can balance the number of mobiles using each Access Channel.

However, it is not clear what are the possible effects if we have more Access Channels? The advantage is clear that more Access Channels means smaller contention slots, thus shorter delays. Nevertheless, it is not clear to us what is the potential disadvantage and *how many Access Channels generate the best system performance?* Since the performance aspect is difficult to be tackled by an analytical approach, we decided to perform experiments for identifying the design trend and possible disadvantage(s). The following Figure 4.7 depicts the preliminary results corresponding to 1, 2, and 4 Access Channels.

The preliminary results demonstrate a very promising results on reducing the T_{update} component. By using 2 Access Channels, the average T_{update} of 5 connections have been reduced from 221 *msec* to 112 *msec* (i.e., 49.3% reduction). The T_{update} for supporting 6 connections has been reduced from 264 *msec* to 134 *msec* (i.e., 49.2% reduction). For the case of 9 connections, T_{update} has been reduced

[2] CSMA stands for Carrier Sense Multiple Access

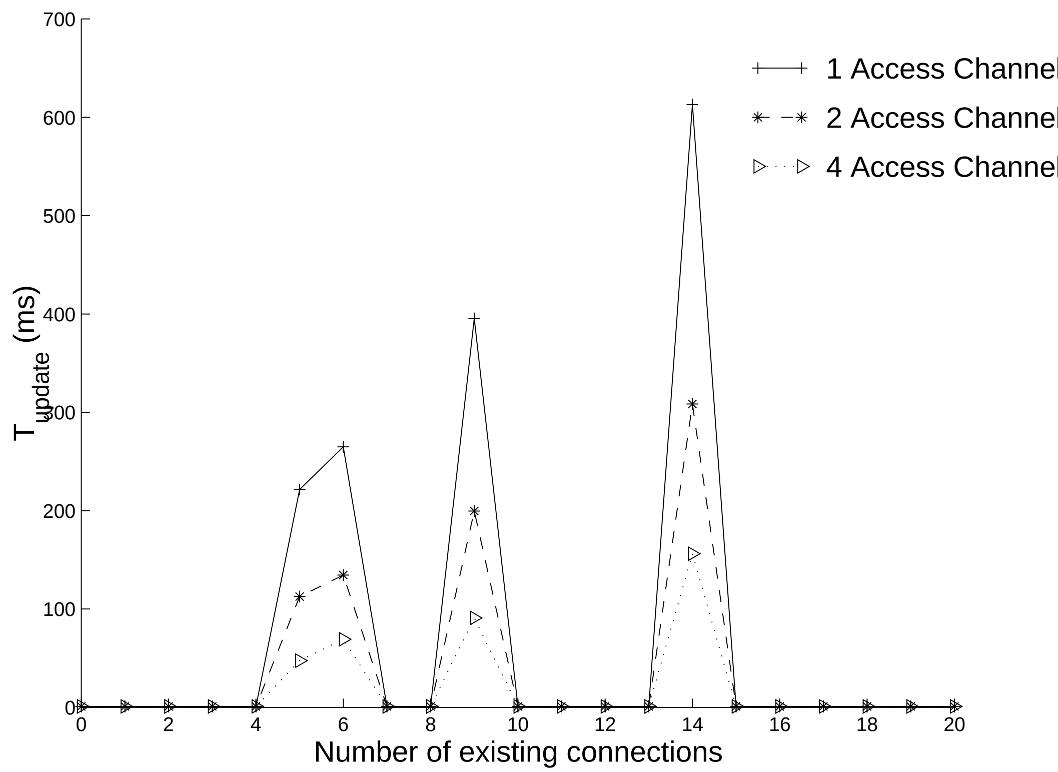

Figure 4.7: Improved T_{update} by using multiple Access Channels.

from 395 *msec* to 199 *msec* (i.e., 49.6% reduction). Finally, by using two Access Channels, T_{update} has been reduced from 612 *msec* to 308 *msec* (i.e., 49.6%) when 14 connections are supported. Therefore, by using two Access Channels, the T_{update} can be reduced 49% stably independent of the number of existing connections.

With 4 Access Channels, the T_{update} component can be further reduced. For supporting (5-, 6-, 9- and 12-) connections, (47.5 *msec*, 69.2 *msec*, 90.99 *msec*, and 156.2 *msec*) are measured correspondingly. Therefore, by increasing the Access Channels from two to four, a further reduction about 48% to 58% of T_{update} is accomplished. Certainly it is interesting to investigate more Access Channels for searching the lowest T_{update} that a system can perform. However there is a significant tradeoff that has been observed by using more Access Channels. Thus, pursuing more than 4 Access Channels perhaps is not recommended at this moment.

Access channels are not used all the time, but when they are used in addition to traffic channels that are already active, the average BER will certainly increase because of the additional interference caused by Access Channels. This increase will occur during the update phase. As we increase the number of channels, we also increase the interference between users. The increased interference will affect the user's quality perception (i.e., BER), thus potentially the quality guarantee will be violated. The following Figure 4.8 depicts the affected BER performance by using 1-, 2- and 4- Access Channels.

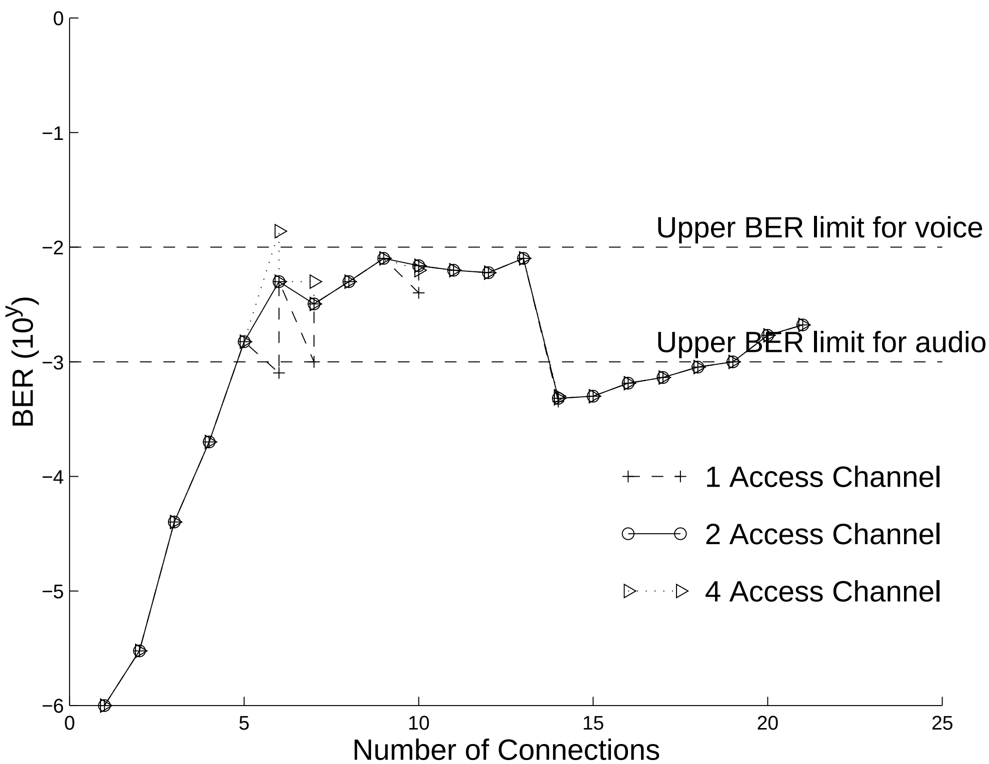

Figure 4.8: The effect in BER with 1, 2, and 4 Access Channels.

When the number of Access Channels is increased from one to two, the BER is affected (i.e., worse quality with larger BER value) when the connections are 6, and 9. Instead of a sharp dropping of the BER when a longer spreading factor, the BER is still increasing (though not sharply). Fortunately, the overall BER performance is still below 10^{-2} guaranteed quality. Therefore, using two Access

Channels proved to be a good method to balance the shorter T_{update} time and BER quality guarantee.

However, when we use 4 Access Channels, the system experiences a sharp increase in BER instead of a drop during a transition. This increase is very large when we switch spreading factor from 32 to 64. The BER requirement for voice is violated during that transition. However, during second transition, the average BER still increases for a short while, but it still stays below the voice limits. The violation of BER guarantee indicates that using 4 Access Channels with spreading factors equal to 32 or 64 is not recommended.

In general, longer spreading factors (e.g., 128 or 160) are more tolerable from having more Access Channels (e.g., 4). For shorter spreading factors (e.g., 32, 64 or 96) the counter effect may be very high. How to balance the two performance metrics between the number of guaranteed users and fast end-to-end setup time seems to be an interesting topic deserving a deep investigation. The preliminary results suggest that the counter effect can potentially reduce the capability of supporting more users. Therefore there is perhaps a need to develop a dynamic scheme that a single Access Channel is used when the spreading factors are small, then gradually increase to multiple Access Channels according to the network load and spreading factors. This topic is beyond the scope of this thesis.

CHAPTER 5
SUMMARY AND CONCLUSION

In this thesis, we proposed a new MAC protocol for wireless CDMA. The protocol can support different traffic classes that have a variety of BER requirements. Since the BER is indirectly proportional to the number of users in a wireless communication system, we designed a protocol with a guaranteed quality maximizing the number of users supported.

The presented protocol admits new calls only if the new connection's as well as all the existing connections' quality can be guaranteed. Our protocol uses a dynamic spreading factor scheme. In order to satisfy the BER requirement, the protocol tries to dynamically change the SF used by connections that may experience a high BER. The connections may alter their SF several times depending on the changes in the number of users in the system.

We have evaluated the BER performance and admission time of the protocol under an increasing number of voice connections. We have also compared our protocol to a regular CDMA protocol that uses a fixed spreading factor. The results show that the DSF protocol provides significant improvement in BER satisfaction compared to CDMA with fixed SF. Even though the number of users increase in the system, the average BER of all connections are maintained below the upper limit for their traffic type. New improvement schemes are proposed and evaluated to reduce the admission delays.

As a future study, we will evaluate the performance of the protocol under more complex conditions (i.e., mixed traffic with streaming video). We will also expand the protocol to cover more than one cell.

APPENDIX
SIMULATION CODE

In this appendix, we include some of our simulation code. We implemented a header file, *protocol.h*, which includes the protocol parameters, base and mobile station class declarations. The following is the content of this file.

```
#include <stdio.h>
#include <stdlib.h>
#include <fstream.h>
#include <iostream.h>
#include <string.h>
#include <time.h>
#include <sys/times.h>
#include <limits.h>
#include <unistd.h>
#include <math.h>
//------constants---------------------
//mobile status
static const int IDLE = 0;
static const int ACTIVE = 1;
static const int WAITING = 2;
static const int RETRY = 3;
//result of a request
static const int ACCEPTED = 1;
static const int DENIED = 0;
```

```
//signals
static const int UPDATE = 27;
static const int OPEN = 28;
static const int CLOSE = 29;
static const int ACK = 30;
//traffic types
static const int VOICE = 0;
static const int AUDIO = 1;
static const int VIDEO = 2;
static const int DATA  = 3;
//miscalenous
static const int NA = 0; //not available
static const int NOPARAM = 0; //no parameter
static const int CHIP_RATE = 4096000; //4.096 Mcps WCDMA
static const int SpreadingFactors[5] = {32,64,96,128,160};
//data rates for voice, audio, video and data.
static const int MinDataRates[4] = {8,16,32,8};
//BER limits for voice, audio, video, and data
static const double UpperBounds[4] = {0.01,0.001,0.0001,0.000001};
//------end of constants--------------
//
//--------data types------------------
struct conn { //data structure for connection information
  int source;
  int dest;
  int type; //traffic type carried on that channel
  int sf; //index for spreading factors
```

```
};
struct PgChMessage { //paging channel message format
  int address; //address 255 is used to broadcast
  int signal;
  int payload[5];
  //various parameters are included in payload
  //e.g. SF, traffic type, result of a request,...
};
struct AccessChMessage { //access channel message format
  int signal;
  char payload[5];
  //various parameters are included in payload
  //e.g. SF, traffic type, source and destination,...
};
//-----end of data types--------------
//
// prototypes
class User;
class BS;
//------class declarationss-------------
//
// Mobile Station class
class User {
  //private variables
  int Status;
  int SF; //spreading factor
  int User_ID;
```

```
  //all users add their access channel messages
  //to the following queue
  Queue *BQ;
  ofstream a_file; //log file for user i
  int MaxRetry; //maximum number of retries in case of a denial
  int retrycount;
  int traffic_type;
  int uptime; //time elapsed during process
  struct AccessChMessage *last_request; //used for retry
  struct AccessChMessage *AccessMsg;
  struct PgChMessage *PgMsg;
  int Marked_APP; //flag, set when the user makes a request
  int Marked_BS;  //flag, set when the base makes a request
  //private functions
  void waitr(); //random wait function
  int Signal_to_BS (); //send request to base station
  int Signal_from_BS (); //send request to mobile station
 public:
  User( Queue *b, int id); //constructor
  int Get_ID(); //returns the user id
  int Process(); //request is processed by mobile
  //request done by user
  int Mark_byAPP(struct AccessChMessage *msg);
  //request done by base station
  int Mark_byBS(struct PgChMessage *msg);
}; //end of user class
```

```
//Base Station class
class BS {
  //private variables
  List *connections; //list of existing connections
  int num_conn; //total number of connections
  User *users[300]; //list of all users in the system
  int num_users; //total number of users
  //base stations receives requests from the following queue
  Queue *BQ;
  //list of messages to be transmitted on paging channel
  List *Messages;
  ofstream a_file; //base station log file
  struct history1 { //stored info for an update
    int traffic_type;
    int sfactor;
  } updated;
  struct history2 { //stored info for an opened connection
    int source;
    int dest;
    int sf;
  } opened;
  int ackcounter; //acknowledgement counter
  //private functions
  //add connection to the list
  void Add2Conn(int source, int dest, int type, int sf);
  int Find_SF (int type); //find spreading factor
  int Calculate_DR ( int factor); //calculate data rate
```

```
  int Get_minDR( int type); //returns minimum data rate for traffic
  int Check_BER(int sf, int type); //checks BER satisfaction
 public:
  BS(Queue *b); //constructor
  //process access request
  int Signal (struct AccessChMessage *request);
  void Process();
  void AddUser( User *u); //add a user to the system
  void CreateConns(int n, int t); //create n connections of type t
}; //end of BS class
//
//---------end of class declarations-----------------------
```

The class functions are implemented in *protocol.cpp*. In this appendix, we include some of the important function implementations. The following is the code implementation of the signaling part in user class. In a sample call procedure, *Mark_byAPP()* is called first. That will set the flag and reserve space for the message. Next, *process()* is called to place the OPEN request in the base station queue (BQ). Base station's *process()* function will retrieve the message from the queue and call the *signal()* function. The admission control protocol is implemented in the *signal()* function.The signal function will execute the code associated with the signal type. After a decision is made in the *signal()* function, it will call the user's *Mark_byBS()* function to send back the result or broadcast the update messages.

```
//signaling
int User::Signal_to_BS()
//function is called when the user makes a request
{
```

```
int i;
switch(AccessMsg->signal) {
case OPEN :
  if (Status != IDLE)
    return -1;
  BQ->enqueue(AccessMsg); //sent to base
  //store the message for retries
  last_request = new AccessChMessage;
  last_request->signal = AccessMsg->signal;
  for (i=0; i<3 ; i++)
    last_request->payload[i] = AccessMsg->payload[i];
  Status = WAITING;
  traffic_type = AccessMsg->payload[2];
  uptime += 8; //time to send an OPEN request to base station
  printf("mobile %d sends open to base\n",User_ID);
  break;
case CLOSE :
  BQ->enqueue(AccessMsg); //sent to base
  Status = IDLE;
  SF = -1;
  traffic_type =-1;
  break;
case UPDATE :
  if (Status == IDLE)
    return -1;
  BQ->enqueue(AccessMsg);
  last_request = new AccessChMessage;
```

```
    last_request->signal = AccessMsg->signal;
    for (i=0; i<5 ; i++)
      last_request->payload[i] = AccessMsg->payload[i];
    Status = WAITING;
    break;
  case ACK:
    BQ->enqueue(AccessMsg);
    break;
  }
  return 0;
} //end of APP signal
int User::Signal_from_BS()
//function called when base station makes a request
{
  struct AccessChMessage *msg;
  switch(PgMsg->signal) {
  case UPDATE :
    if ((Status == IDLE) || (Status == WAITING))
      return -1;
    //not for this user
    if (traffic_type != PgMsg->payload[2])
      return -1;
    printf("mobile %d received update and sending back ack\n",
            User_ID);
    SF = PgMsg->payload[3];
    msg = new AccessChMessage;
    msg->signal = ACK;
```

```
    msg->payload[0] = User_ID;
    msg->payload[1] = UPDATE;
    msg->payload[2] = ACCEPTED;
    //send back ACK
    BQ->enqueue(msg);
    break;
  case CLOSE:
    Status = IDLE;
    traffic_type = -1;
    SF = -1;
    msg = new AccessChMessage;
    msg->signal = ACK;
    msg->payload[0] = User_ID;
    msg->payload[1] = CLOSE;
    msg->payload[2] = ACCEPTED;
    //send back ACK
    BQ->enqueue(msg);
    break;
  case OPEN:
    if (Status != IDLE) { //station is busy, cannot accept request
      msg = new AccessChMessage;
      msg->signal = ACK;
      msg->payload[0] = User_ID;
      msg->payload[1] = OPEN;
      msg->payload[2] = DENIED;
      //send back ACK
      BQ->enqueue(msg);
```

```
    uptime += 8; //time to send back an ACK
    return 0;
  }
  //request accepted
  uptime += 8; //sending back ack
  SF = PgMsg->payload[3];
  traffic_type = PgMsg->payload[2];
  Status = ACTIVE;
  msg = new AccessChMessage;
  msg->signal = ACK;
  msg->payload[0] = User_ID;
  msg->payload[1] = OPEN;
  msg->payload[2] = ACCEPTED;
  //send back ACK
  printf("mobile %d received open from base
          and sending back ack\n",User_ID);
  BQ->enqueue(msg);
  break;
case ACK: //if the user is in waiting
  if (Status != WAITING)
    return -1; //error-not in waiting status
  if (PgMsg->payload[0] == DENIED)
    waitr(); //wait for a random time
  else { //OPEN request accepted, update SF and status
    uptime += 4; //time to update parameters
    Status = ACTIVE;
    SF = PgMsg->payload[1];
```

```
      printf("mobile %d received ack to open
              and sf is %d\n",User_ID,SpreadingFactors[SF]);
    }
    break;
  }
  return 0;
} //end of BS signal
//a request is done by application
int User::Mark_byAPP(AccessChMessage *r){
  if (Marked_APP == 1)
    return -1;
  Marked_APP = 1;
  AccessMsg = r;
  return 0;
}
//a request is done by base station
int User::Mark_byBS(PgChMessage *r) {
  if (Marked_BS == 1)
    return -1;
  Marked_BS = 1;
  PgMsg = r;
  return 0;
}
int User::Process(){
  uptime = 0;
  if (Marked_APP == 1)
    Signal_to_BS();
```

```
  if (Marked_BS == 1)
    Signal_from_BS();
  if ((Status == RETRY) && (retrycount > 0))
    //if waiting to retry, decrement the waiting count
    retrycount--;
  else if ((Status == RETRY) && (retrycount == 0)) { //do the retry
    AccessMsg = last_request;
    Signal_to_BS();
  }
  printf("user process time %d\n",uptime);
  return 0;
}
```

The following is the code implementation of the signaling part in base station class.

```
//signaling
int BS::Signal (struct AccessChMessage *request) {
  int f, sfactor, datarate;
  struct PgChMessage *pg, *pgbroadcast;
  struct conn *existing, *c;
  Queue store;
  int userid = request->payload[0];
  int traffic_type, newf;
  double retr;
  int first;
  switch(request->signal) {
  case OPEN:
    pg = new PgChMessage;
```

```
//check if there is already a connection open for dest
if ( connections->IsOpen(request->payload[1]) == 1) {
  //the dest is busy
  pg->address = userid;
  pg->signal = ACK;
  pg->payload[0] = DENIED;
  pg->payload[1] = -1;
  printf("request denied because it is busy\n");
  users[userid]->Mark_byBS(pg);
  return 0;
}
//the dest is not busy
ptime += 2; //time to find SF for new connection
num_conn = num_conn + 2;
sfactor = Find_SF(request->payload[2]);
datarate = Calculate_DR (sfactor);
if ( datarate < Get_minDR(request->payload[2]) ) {
  //cannot satisfy the connection
  num_conn= num_conn - 2;
  pg->address = userid;
  pg->signal = ACK;
  pg->payload[0] = DENIED;
  pg->payload[1] = -1;
  printf("cannot be satisfied because data rate small\n");
  users[userid]->Mark_byBS(pg); //reject the open request
  return 0;
}
```

```
printf("new connection's spreading factor =
        %d\n\t",SpreadingFactors[sfactor]);
f = -1;
//check existing connections
updated.sfactor = -1;
first = 1;
connections->InitNext();
while (( existing = connections->Next()) != NULL) {
  ptime += 1;
  if (Check_BER(existing->sf, existing->type)== DENIED) {
    //connection cannot be satisfied
    if (first == 1) { //if it is first connection of type i
      ptime += 2; //takes more time since it calculates SF
      first = 0;
    }
    else
      ptime += 1; //no need to find SF for that connection
    f = Find_SF(existing->type);
    datarate = Calculate_DR(f);
    if ( datarate < Get_minDR(existing->type) ) {
      num_conn= num_conn - 2;  // deny the new connection
      pg->address = userid;
      pg->signal = ACK;
      pg->payload[0] = DENIED;
      pg->payload[1] = -1;
      users[userid]->Mark_byBS(pg); //reject the open request
      return 0;
```

```
      }
      updated.traffic_type = VOICE;
      updated.sfactor = f;
    }
  }
  if (f != -1)
  opened.source = userid;
  opened.dest = request->payload[1];
  opened.sf = sfactor;
  //all existing connections are satisfied
  //notify destination
  ptime += 4; //time to send request to dest
  pg->address = request->payload[1];
  pg->signal = OPEN;
  pg->payload[0] = request->payload[0];
  pg->payload[1] = request->payload[1];
  pg->payload[2] = request->payload[2];
  pg->payload[3] = sfactor;
  users[request->payload[1]]->Mark_byBS(pg);
  break;
/*---------------------------------------------
      Procedures for CLOSE and UPDATE
     requests are also implemented here
---------------------------------------------*/
case ACK:
  if (request->payload[1] == UPDATE) {
    ackcounter++;  //receiving ACKs for UPDATE broadcast
```

```
  if (ackcounter >= num_conn) { //all acks are received
    pg = new PgChMessage;  //activate the connection
    pg->address = request->payload[1];
    pg->signal = ACK;
    pg->payload[0] = ACCEPTED;
    pg->payload[1] = opened.sf;
    users[opened.source]->Mark_byBS(pg);
    //add to connections list
    c = new conn;
    c->source = opened.source;
    c->dest = opened.dest;
    c->type = VOICE;
    c->sf = opened.sf;
    printf("adding new connection\n");
    connections->Add(c);
  }
}
else if (request->payload[1] == OPEN) {
  if (request->payload[2] == ACCEPTED) {
    //the destination accepts the open request
    if (updated.sfactor > -1) {
      //broadcast an UPDATE message
      pgbroadcast = new PgChMessage;
      pgbroadcast->address = 255;
      pgbroadcast->signal = UPDATE;
      pgbroadcast->payload[2] = updated.traffic_type;
      pgbroadcast->payload[3] = updated.sfactor;
```

```
  connections->InitNext();
  //2 access channels are use
  retr = 2.71828 * ceil((num_conn-2)/2);
  ptime+= retr*8; //time spent during contention period
  ptime+= 4; //time to send broadcast
  while (( existing = connections->Next()) != NULL) {
    printf("sending update to %d and %d\n",
            existing->source,existing->dest);
    //send update to source
    users[existing->source]->Mark_byBS(pgbroadcast);
    //send update to dest
    users[existing->dest]->Mark_byBS(pgbroadcast);
  }
}
else {
  //no broadcast is necessary
  ptime += 6; //send back ack for open + add new conn
  pg = new PgChMessage;
  pg->address = request->payload[1];
  pg->signal = ACK;
  pg->payload[0] = ACCEPTED;
  pg->payload[1] = opened.sf;
  users[opened.source]->Mark_byBS(pg);
  //add to connections list
  c = new conn;
  c->source = opened.source;
  c->dest = opened.dest;
```

```
          c->type = VOICE;
          c->sf = opened.sf;
          connections->Add(c);
        }
      }
      else {
        //the destination denied the OPEN request
        pg = new PgChMessage;
        pg->address = request->payload[1];
        pg->signal = ACK;
        pg->payload[0] = DENIED;
        pg->payload[1] = -1;
        users[opened.source]->Mark_byBS(pg);
        num_conn = num_conn - 2;
        //the connection cannot be established
        connections->Delete(opened.source);
      }
    }
    break;
  }
}
void BS::Process() {
  struct AccessChMessage *r;
  ptime =0;
  while ((r=BQ->dequeue()) != NULL) {
    Signal(r); }
  printf("total time = %f msec\n",ptime); }
```

REFERENCES

[1] 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, no. 2, April 1999, pp. 146-157.

[2] I. F. Akyildiz, J. McNair, L. Carrasco, R. Puigjaner, and Y. Yesha, "Medium Access Control Protocols for Multimedia Traffic in Wireless Networks," *IEEE Network,* July/August 1999, pp. 39-47.

[3] P. Chang and C. Lin, "Design of Spread Spectrum Multi-code CDMA Transport Architecture for Multimedia Services," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 1, January 2000, pp. 99-111.

[4] S. Choi and D. Cho, "Capacity Evaluation of Forward Link in a CDMA System Supporting High Data-rate Service," *in Proc. 2000 IEEE GLOBECOM,* San Francisco, pp. 123-127.

[5] S. Choi and K. G. Shin, "Uplink CDMA Systems with Diverse QoS Guarantees for Heterogeneous Traffic," *in Proc. MOBICOM'97,* Budapest, pp. 120-130.

[6] S. Choi and K. G. Shin, "A Unified Wireless LAN Architecture for Real-Time and Non-Real-Time Communication Services," *IEEE/ACM Transactions on Networking,* vol. 8, no. 1, February 2000, pp. 44-59.

[7] P. K. Frenger, P. Orten, and T. Ottosson, "Code-Spread CDMA with Interference Cancellation," *IEEE Journal on Selected Areas in Communications,* vol. 17, no. 12, December 1999, pp. 2090-2095.

[8] H. Fukumasa, R. Kohno, and H. Imai, "Design of Pseudo-noise Sequences with Good Odd and Even Correlation Properties for DS/CDMA," *IEEE Journal on Selected Areas in Communications,* vol. 12, no. 5, June 1994, pp. 828-836.

[9] E. Geraniotis and J. Wu, "The Probability of Multiple Correct Packet Receptions in Direct-Sequence Spread-Spectrum Networks," *IEEE Journal on Selected Areas in Communications,* vol. 12, no. 5, June 1994, pp. 871-884.

[10] L. Harte, *CDMA IS-95 for Cellular and PCS,* New York: McGraw-Hill, 1999.

[11] C. I, C. A. Webb III, H. C. Huang, S. Brink, S. Nanda, and R. D. Gitlin, "IS-95 Enhancements for Multimedia Services," *Bell Labs Technical Journal,* vol. 1, no. 2, Autumn 1996, pp. 60-67.

[12] J. S. Lee and L. E. Miller, *CDMA Systems Engineering Handbook,* Boston, Mass.: Artech House, 1998.

[13] M. Lops, G. Ricci, and A. M. Tulino, "Narrow-Band-Interference Suppression in Multiuser CDMA Systems," *IEEE Transactions on Communications,* vol. 46, no. 9, September 1998, pp. 1163-1175.

[14] T. Nagaosa and T. Hasegawa, "Code Assignment and the Multi-code Sense Scheme in an Inter-Vehicle CDMA Communication Network," *IEICE Trans. Fundamentals,* vol. E81-A, no. 11, November 1998, pp. 2327-2333.

[15] M. Naraghi-Pour and H. Liu, "Integrated Voice-Data Transmission in CDMA Packet PCN's," *in Proc. 2000 IEEE ICC,* New Orleans, pp. 1085-1089.

[16] S. Oh and K. M. Wasserman, "Dynamic Spreading Gain Control in Multiservice CDMA Networks," *IEEE Journal on Selected Areas in Communications,* vol. 17, no. 5, May 1999, pp. 918-927.

[17] L. Zhuge and V. Li, "Interference Estimation for Admission Control in Multiservice DS-CDMA Cellular Systems," *in Proc. 2000 IEEE GLOBECOM,* San Francisco, pp. 1509-1514.

BIOGRAPHICAL SKETCH

Mehmet Ali Elicin was born on October 22nd, 1976, in Balikesir, Turkey. He received his Bachelor of Science degree from Bogazici University, Istanbul, in June 1999, majoring in computer engineering. He joined the University of Florida in August 1999 and started to pursue a master's degree in computer and information science and engineering. He worked as a teaching assistant during his master's degree.

His research interests include computer networks, wireless communication and multimedia communication.