

**ANALYSIS AND PERFORMANCE EVALUATION
OF AN EFFICIENT PROTOCOL FOR THE CDPD
FORWARD CHANNEL**

by

XU (JAKE) ZHANG

B. Eng., The Beijing Broadcast Institute, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

September 2001

© Xu (Jake) Zhang, 2001

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

Department of ELECTRICAL & COMPUTER ENGINEERING

The University of British Columbia
Vancouver, Canada

Date Sept. 28, 2001

Abstract

The purpose of this thesis is to analyze and evaluate the performance of the Cellular Digital Packet Data (CDPD) forward channel at the Medium Access Control (MAC) layer, and propose an efficient protocol to improve the MAC layer throughput.

First we study the MAC layer by several performance measures, such as the Block Error Rate (BLER), Packet Error Rate (PER), and the throughput. According to our studies on the forward channel MAC layer structure, we analyze the relationship among the three performance measures, and between them and the channel Bit Error Rate (BER), in the presence of the Additive White Gaussian Noise (AWGN) channel. For comparison purposes, computer simulations have also been conducted with the AWGN channel, and land-mobile Rayleigh fading channel, with the employment of both a coherent and a 1-bit differential GMSK receiver.

As a result of this study, a source of performance inefficiency, namely, the Correct but Unusable Data (CUD), is identified. We argue that the MAC layer throughput is affected not only by the error performance of the physical layer, but also by the MAC layer block segmentation, which is the main cause of the CUD. The percentage of CUD peaks to about 45% regardless the channel conditions. In order to eliminate this inefficiency we propose a MAC-ARQ protocol which performs ARQ operation at the MAC layer. Verified by computer simulations, this protocol effectively improves the throughput at certain channel conditions, while slightly degrades the performance at other conditions. Finally, in order to maintain maximum throughput over all channel conditions, we propose an adaptive scheme which automatically switches back and forth between the conventional ARQ protocol and the MAC-ARQ protocol according to the channel conditions.

Table of Contents

Abstract	ii
List of Figures	vi
List of Tables	viii
List of Abbreviations	ix
Acknowledgment	xi
Chapter 1 Introduction	1
1.1 CDPD Overview [2].....	2
1.2 Performance Considerations	4
1.3 Research Objectives of the Thesis	8
1.4 Thesis Organization.....	9
Chapter 2 CDPD System Model	11
2.1 Introduction.....	11
2.2 PHY Layer Configuration	13
2.2.1 GMSK Transmitter	13
2.2.2 Communication Channel Model.....	15
2.2.3 GMSK Receivers	18
2.3 MAC Layer Configuration.....	21
2.3.1 MAC Layer Configuration at the MDBS	22
2.3.2 Packet Reconstruction at the M-ES.....	24
2.3.3 Medium Access Algorithm	26
2.4 Logic Link Layer and ARQ Procedure	26

2.4.1	LLC Layer Frame Structure	27
2.4.2	ARQ Procedure	28
2.5	Computer Simulation Model and Parameters	29
2.5.1	MDBS Model.....	29
2.5.2	Communication Channel Model.....	30
2.5.3	M-ES Model	31
2.6	Summary	31
Chapter 3	Performance Analysis and Evaluation	33
3.1	Introduction.....	33
3.2	Correct but Unusable Data	33
3.3	Block and Packet Error Performance	36
3.4	Throughput Performance	40
3.5	Numerical Results and Discussion	42
3.5.1	BLERs and PERs	42
3.5.2	CUD	51
3.5.3	Throughput	53
3.6	Summary	56
Chapter 4	MAC-ARQ Protocol	57
4.1	Introduction.....	57
4.2	Description and Performance Evaluation of the MAC-ARQ Protocol	57
4.3	The Adaptive Scheme	63
4.4	Summary	68
Chapter 5	Conclusion and Some Suggestions for Future Research	69

5.1	Conclusions	69
5.2	Suggestions for Future Research	70
5.2.1	Effect of the MAC-ARQ protocol to the reverse channel	70
5.2.2	Revoke the ARQ operation at the data link layer for the MAC-ARQ scheme	70
5.2.3	Soft threshold for the adaptive scheme	70
5.2.4	Performance study for similar applications other than CDPD	71
5.2.5	Diversity reception	71
5.2.6	More advanced coding techniques	71
	References	72
	Appendix A. Program Listings	76

List of Figures

Figure 1.1	General CDPD system architecture [3].....	3
Figure 2.1	OSI reference model.....	12
Figure 2.2	M-ES protocol stack.....	12
Figure 2.3	Block diagram of GMSK transmitter.....	14
Figure 2.4	(a) Pulse response of Gaussian lowpass filter. (b) Corresponding power spectral of GMSK signals [16].....	16
Figure 2.5	Block diagram of communication channel model.....	18
Figure 2.6	Block diagram of coherent GMSK detector.....	19
Figure 2.7	Block diagram of GMSK 1-bit conventional differential detector	20
Figure 2.8	Data transmission procedures in the forward channel.....	23
Figure 2.9	Flow chart for (a) procedure of FCTS generation at the MDBS; (b) procedure of LPDU reconstruction at the M-ES	25
Figure 2.10	Structure of the LLC frames.....	27
Figure 2.11	Block diagram of computer simulation model	29
Figure 3.1	Generation of correct but unusable data (CUD).....	34
Figure 3.2	Percentage of CUD in the AWGN channel with coherent detection.....	35
Figure 3.3	Probability of a packet in error (P_P) versus Probability of a block in error (P_B).....	39
Figure 3.4	BLER and PER in the AWGN channel.....	43
Figure 3.5	BLER and PER with differential detection in the Rayleigh fading channel with $f_D = 20$ Hz	45
Figure 3.6	BLER and PER with differential detection in the Rayleigh fading channel with $f_D = 40$ Hz	46
Figure 3.7	BLER and PER with differential detection in the Rayleigh fading channel with $f_D = 80$ Hz	47

Figure 3.8	Comparison of BLERs and PERs with different Doppler shift frequencies with differential detection in the Rayleigh fading channel. (a) BLER. (b) PER.....	49
Figure 3.9	BER with error correction coding in the Rayleigh fading channel.	50
Figure 3.10	Percentage of CUD with differential detection in the Rayleigh fading channel.	52
Figure 3.11	Throughput versus BER with coherent detection in the AWGN channel, where the packets are transmitted in their default lengths.	54
Figure 3.12	Throughput versus E_b/N_0 in the Rayleigh fading channel.	55
Figure 4.1	Data transmission procedures in the forward channel.	58
Figure 4.2	Throughput versus BER with coherent detection in the AWGN channel, where the packets are transmitted in their default lengths.	61
Figure 4.3	Throughput versus E_b/N_0 with differential detection in Rayleigh fading channel, where the packets are transmitted in their random lengths.	62
Figure 4.4	Throughput versus BER with differential detection in Rayleigh fading channel ($f_D=20\text{Hz}$), where the packets are transmitted in their random lengths.	64
Figure 4.5	Throughput versus BER with differential detection in Rayleigh fading channel ($f_D=40\text{Hz}$), where the packets are transmitted in their random lengths.	65
Figure 4.6	Throughput versus BER with differential detection in Rayleigh fading channel ($f_D=80\text{Hz}$), where the packets are transmitted in their random lengths.	66

List of Tables

Table 3.1	The 90% confidence interval for each BLER	44
Table 4.1	Observed BER thresholds for various Doppler shifts.....	63

List of Abbreviations

AMPS	Advanced Mobile Phone Service
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BLER	Block Error Rate
CD	Collision Detection
CDPD	Cellular Digital Packet Data
CFDBS	Continuous Frame Data Bit Stream
CLNP	Connectionless Network Protocol
CPM	Continuous Phase Modulation
CSMA	Carrier Sense Multiple Access
CUD	Correct but Unusable Data
DSMA	Digital Sense Multiple Accessing
ES	End System
FCTS	Forward Channel Transmission Sequence
FEC	Forward Error Correction
F-ES	Fixed End System
FM	Frequency Modulation
GBPF	Gaussian Bandpass Filter
GLPF	Gaussian Low Pass Filter
GMSK	Gaussian Minimum Shift Keying
GPRS	General Packet Radio Service
GSM	Groupe Speciale Mobile
IP	Internet Protocol
IS	Intermediate System
ISI	Inter-Symbol Interference
LLC	Logic Link Control
LP	Large Packet
LPDU	Link Protocol Data Unit

MAC	Medium Access Control
MDBS	Mobile Data Base Station
MD-IS	Mobile Data Intermediate System
MDLP	Mobile Data Link Protocol
M-ES	Mobile End System
MP	Medium Packet
NAK	Negative Acknowledgment
NRZ	Nonreturn to Zero
OSI	Open System Interconnection
PER	Packet Error Rate
PHY	Physical Layer
RF	Radio Frequency
RLC	Radio Link Control
RS	Reed-Solomon
RS-DB	Reed-Solomon Data Block
SNDCCP	Subnetwork Dependent Convergence Protocol
SP	Small Packet
S-REJ	Selective-Reject
WWW	World Wide Web

Acknowledgments

This thesis is dedicated to my father, Hangfu Zhang, my mother, Lanying Guo, and my wife Man Li, who have been giving me constant, unconditional support during my studies. I am enormously grateful to my supervisor, Dr. P. T. Mathiopoulos, for his constant encouragement and continual guidance. Without his invaluable suggestions and support, the completion of this thesis would not have been possible. Also I would like to thank Dr. A. Salkintzis and my friend Hong Nie for their useful discussions and suggestions on the thesis work.

Chapter 1 Introduction

Cellular Digital Packet Data (CDPD) is a mobile data technology that permits subordinate packet data operation on the spectrum assigned to a telephone cellular network, such as the Advanced Mobile Phone Service (AMPS). It is considered as a very promising mobile data technology featuring many important benefits [1]. Among many other wireless communication standards, CDPD provides satisfactory performance with comparatively less complexity. For this reason, CDPD technology has been deployed and supported by many major wireless data operators particularly in North America, including GTE Wireless, AT&T Wireless, Bell Atlantic Mobile, Ameritech, and McCaw Cellular.

CDPD was firstly designed to accommodate short data transmissions during the interval between the talk spurts of a voice network. It is ideally suited for applications where small amounts of data need to be transferred from remote locations to a central system. Typical examples of its applications include telemetry, vehicle tracking, and personal messaging, etc. However, due to the phenomenal growth of the wireless communication market and the popularity of the Internet, the demand for mobile data exchange, e.g. mobile internet access, has been increasing considerably. Particularly, the potential of applications and markets built around the World Wide Web (WWW) is enormous. Because these WWW applications usually involve rich multimedia data downloading, it is important for CDPD to support these applications effectively, especially in the forward channel, i.e. the channel from the central system to the mobile user. CDPD supports 19.2-kbps raw data rate at the physical layer, which, at best, allows a mobile user to be able to obtain throughput performance similar to that of a 14.4-kbps wireline modem [1]. Therefore, for applications which only involve a moderate amount of graphics, CDPD is still a

practical and appropriate solution. In order to effectively maintain and improve the performance, higher transmission efficiency is required on the CDPD forward channel. Consequently it is important to investigate the characteristics of this channel in order to determine its capabilities, to reveal its possible deficiencies and to propose potential improvements. In this thesis, the performance of CDPD forward channel is studied; a source of data transmission deficiency is identified, and a new protocol, which eliminates this degradation, is proposed, analyzed, and its performance evaluated.

1.1 CDPD Overview [2]

The important subsystems of a CDPD network are the End Systems (ESs) and the Intermediate Systems (ISs), as shown in Fig. 1.1. The ESs represent the actual physical and logical end nodes that exchange information, while the ISs represent the CDPD infrastructure subsystems that store, forward and route the information. In Internet terminology, the ESs are known as hosts and ISs are known as routers.

There are two kinds of ESs: The Mobile End System (M-ES), which is a device used by a mobile subscriber to access the CDPD network over the wireless interface, and the Fixed End System (F-ES), which is a common host, server or gateway with fixed connection to the CDPD backbone and provides access to specific applications and data.

On the other hand, there are two kinds of ISs: A “generic” IS, which is simply a router (in most cases, an Internet Protocol (IP) router) with no knowledge of CDPD and mobility issues, and Mobile Data Intermediate System (MD-IS), which is a specialized IS that routes messages based on its knowledge of the current location of M-ESs. The MD-IS is a set of hardware components and software functions that provide switching, accounting, registration, authentication, encryption

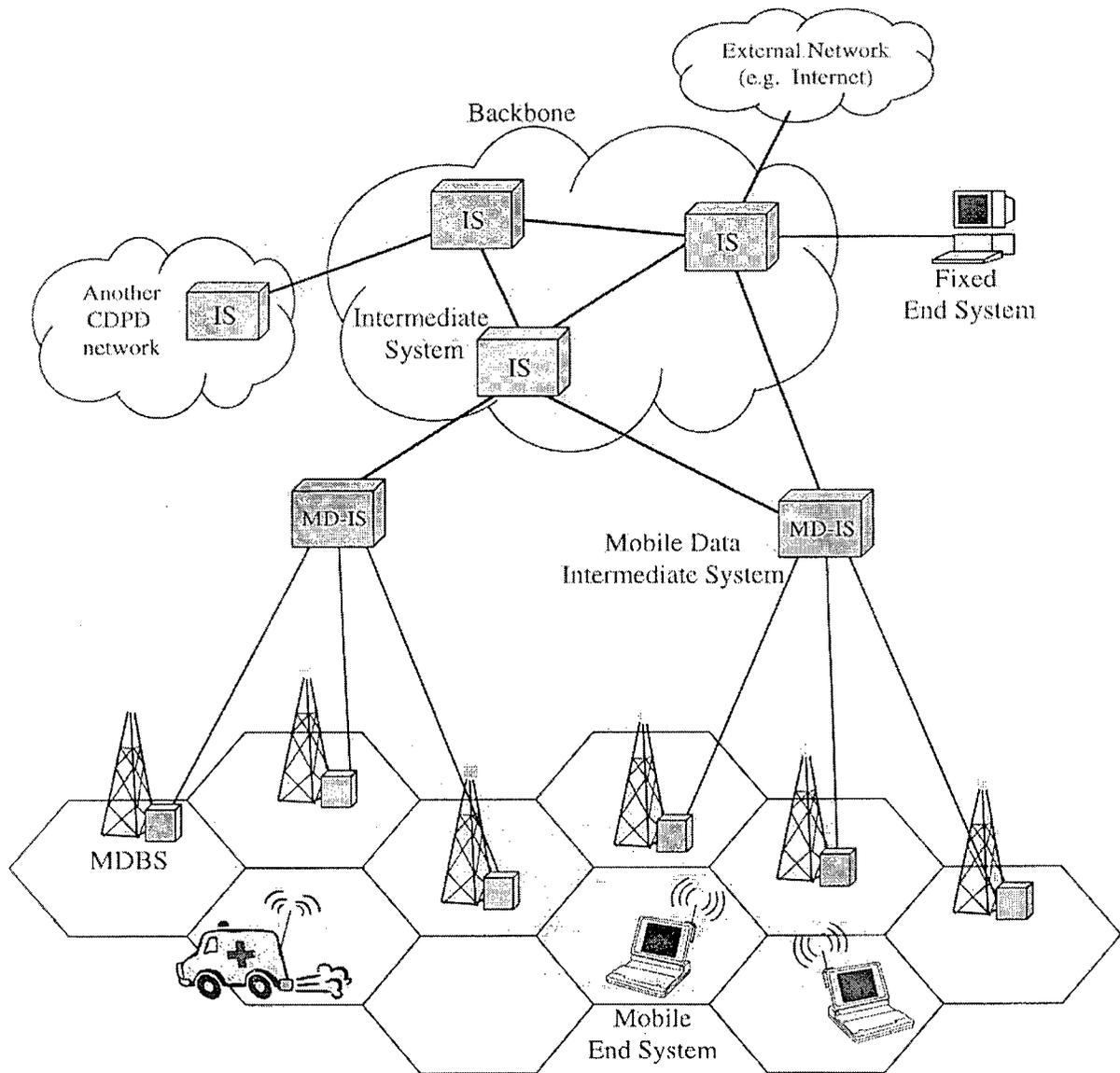


Figure 1.1 General CDPD system architecture [3]

and mobility management functions. Besides the ESs and ISs, there is another subsystem, the Mobile Data Base Station (MDBS), which is analogous to a common cellular base station. MDBS creates and manages an air interface between the M-ES and the CDPD backbone. It performs no networking functions but rather relays data link information between a number of M-ESs and their serving MD-IS, which is a data-link functional subsystem. It also performs radio resource

management procedures such as channel hopping¹.

The CDPD backbone provides connectionless transport services, or “datagram” services, where the network individually routes packets based upon the destination address each packet carries and the knowledge of the current network topology. For example, it supports routing of both IP packets (with the Internet protocol) and the CLNP packets (with the Connectionless Network Protocol) in the Open System Interconnection (OSI) reference model [2].

Although the CDPD was originally designed to accomplish data transmission over existing voice communication channels, it can also be implemented over a separated channel, which is permanently dedicated to the CDPD data transmissions. This is referred to as a dedicated channel, while in contrast, in the former case it is referred to as a nondedicated channel, where the CDPD shares a common channel pool with the underlying cellular voice system [3]. In this case, the CDPD has to release its forward channel whenever this channel is selected for voice transmission.

1.2 Performance Considerations

The increasing of the CDPD popularity has attracted significant growth of research activities on CDPD performance [4-11]. In general, the performance of the CDPD can be evaluated by the efficiency of data transmission from the source node to the destination node. Due to limited frequency bandwidth and data rate, it is easy to see that the bottle neck of the performance is the wireless interface, i.e. the airlink between the MDBS and the M-ESs. As it will be explained with details in the next chapter, the MDBS functions at and below the medium access control (MAC)

¹ Channel hopping is required for a nondedicated channel where CDPD overlays the existing cellular voice network [1].

layer, and relays information from the logic link control (LLC) layer between the M-ES and the M-IS. Hence, it would be beneficial to us by conducting a performance study from the MAC layer point of view. The study will provide us with an idea of how the system behaves over the airlink. Therefore, by identifying and eliminating any possible inefficiencies, the overall system performance can be improved.

Firstly, let us consider the main causes of performance degradations in the CDPD airlink. They usually fall into the following three categories:

i) Impact of the underlying voice network

This applies only to the nondedicated channel system, where the CDPD overlays on top of the existing voice channel. It “borrows” the voice channel when it is available and hop to another free one before the voice connection is set up in this channel. Saha and Kay[5] as well as Budka [8] have investigated the effects of CDPD on the voice channel and that of the voice channel to the CDPD. They have concluded that the CDPD can provide efficient data transmission without substantial degradations to the voice quality.

ii) Sharing of the reverse channel

The access to the reverse channel is shared by the M-ESs within the same cell. Therefore, the CDPD employs an algorithm at the MAC layer, namely, the Digital Sense Multiple Accessing with Collision Detection (DSMA/CD), to regulate the access of the M-ESs and to avoid collision. In the past, several studies, including [5], [10], and [11], have been carried out on the performance with DSMA/CD, mainly focusing on their time delay performance, or the delay throughput.

iii) Physical channel impairments

A noisy mobile communication channel may cause severe distortion to the transmitted signal waveform, yielding errors to a large number of bits. To combat this, coding techniques can be employed, such as the Forward Error Correction (FEC) codes, which performs error checking and correction at the receiver end [33]. Also the type of the signal detector employed at the receiver can affect the performance significantly. It should be chosen so that it is robust to the particular type of channel impairments. For example, in the CDPD, where the Gaussian Minimum Shift-Keying (GMSK) modulation scheme is employed, a differential detector may have better error performance than a coherent detector in a mobile communication channel [28]. There are also other approaches on improving the performance at the physical layer such as to employ diversity or smart antenna technologies at the mobile receiver [4].

As discussed, among the performance studies carried on CDPD so far, only a few address the performance at the MAC layer [5][10][11]. To our best knowledge, there is even no studies focused particularly on the forward channel. Therefore, in this thesis we concentrate on the performance of the forward channel MAC layer. Specifically, we will be investigating a number of performance measures that can be considered for the evaluation of the forward channel performance, including the average Block Error Rate (BLER), the average Packet Error Rate (PER), and the MAC layer throughput. The BLER is the probability of erroneous blocks encoded with error correction codes at the MAC layer. The PER is the probability of erroneous packets generated from the layer above the MAC layer. The throughput represents the transmission efficiency of the actual information at the MAC layer. From the user point of view, these performance measures are more important than the equivalent physical layer measures, such as the average Bit Error Rate (BER), since they demonstrate more closely what the user perceives in terms of performance. However, for comparison purposes, the BERs in the physical channel have

also been included in this thesis. Although several studies have been carried out on the estimation of the BLER in mobile communication channels (e.g. [12], [13], and [14]), however, researches on the relations between the BLER and PER are relatively rare. Hence, in this thesis we will investigate the performance of BLER and PER and their relations. We also identify a source of performance degradation, which we term as “the correct but unusable data” (CUD). It is caused by the variety of the packet lengths and the alignment between the packets and the Reed-Solomon (RS) coded blocks at the MAC layer.

To perform a realistic study it was necessary to make several assumptions about the physical layer, such as the type of the communication channel and the type of the detector used at the receiver to detect the GMSK signal. As it will be explained later, we have considered in our computer simulation studies the communication channel to be a narrowband land-mobile Rayleigh fading channel with Additive White Gaussian Noise (AWGN). Furthermore, we have investigated the performance of two types of receiver structures, namely, the coherent receiver and the differential receiver.

During our performance studies, we have identified a source of performance degradation that causes the receiver to discard data that is not corrupted by the channel impairments. To mitigate this performance degradation we have proposed a new protocol, namely, the MAC-ARQ protocol, which performs the auto-repeat request (ARQ) operations at the MAC layer instead of the LLC layer. There are similar approaches applied or proposed to other applications [39][40], where the auto retransmission mechanism is eliminated in the data link operation and implemented in the MAC layer. In this thesis, on the other hand, our purpose is to minimize structural modifications to the well-defined CDPD standard, while improve the performance as

much as possible. Therefore, we leave the conventional ARQ mechanism in the data link layer and propose another selective ARQ operation in the MAC layer. Rather than implementation details, we concentrate on the analysis and evaluation of the performance differences between the conventional scheme and the proposed approach. We demonstrate considerable improvements to the performance under certain channel conditions. Considering the trade-off it also brings up, we discuss an adaptive scheme that dynamically enables/disables the activation of the MAC-ARQ scheme according to the channel conditions.

1.3 Research Objectives of the Thesis

Based upon the above discussion, it is worth and of interest to study the forward channel MAC layer performance, identifying, and eliminating (or mitigating) possible inefficiencies, hence improve the overall system performance. In this respect the main research objectives of this thesis can be identified as follows.

1. To analyze the relationship among the MAC layer BLER, the PER, and the throughput at the forward channel, and the relationship between these measurements and the channel BER. Additionally to evaluate analytically and by means of computer simulations, these performance measures with various GMSK receiver types, in the presence of both AWGN channel and land-mobile Rayleigh fading channel.
2. To eliminate a source of performance degradation identified as the CUD, and hence improve the forward channel performance in certain conditions by a new MAC-ARQ protocol.
3. To propose and evaluate an adaptive ARQ scheme to dynamically choose the appropriate ARQ operation with better throughput, and hence improve the overall system perfor-

mance.

1.4 Thesis Organization

Including this introductory chapter, this thesis consists of five Chapters and one Appendix. Its organization is as follows:

Chapter 2 describes the CDPD system model. It starts with an introduction in Section 2.1. Afterwards, a detailed description of the physical layer configuration is presented in Section 2.2. Then the configuration of MAC layer is detailed in Section 2.3, followed by the description of the LLC layer configuration and the ARQ operation in Section 2.4. The computer simulation model and parameters are presented in Section 2.5. The chapter is concluded with a summary in Section 2.6.

Chapter 3 evaluates the forward channel performance in terms of the measures discussed above, and introduces the performance degradation source. A brief introduction is presented in Section 3.1. The description of the degradation source is detailed in Section 3.2. Then the relations between the BLER and PER are analyzed in Section 3.3, followed by the derivation of the throughput expression in Section 3.4. Numerical results of the computer simulations and discussions are presented in Section 3.5, and a summary of the chapter appears in Section 3.6.

Chapter 4 provides descriptions of the proposed MAC-ARQ protocol. After a brief introduction in Section 4.1, the configuration of this protocol is presented in Section 4.2. The throughput evaluation of the proposed protocol is presented in Section 4.3. The adaptive scheme will be discussed in Section 4.4, and the summary of this chapter is in Section 4.5.

Chapter 5 presents conclusions of this thesis and some suggestions for future research.

Appendix A contains program listings for the computer simulations, including the GMSK transmitter, the 1-bit differential detector, and the Rayleigh fading simulator.

Chapter 2 CDPD System Model

2.1 Introduction

The CDPD standard supports both the internet protocol stack and the Open System Interconnection (OSI) reference model developed by the International Standards Organization [15]. As depicted in Fig. 2.1, it consists of seven layers, namely, the physical layer, the data link layer, the network layer, the transport layer, the session layer, the presentation layer and the application layer. In order to assure the compatibility of CDPD networks with existing applications software, the protocols above the network layer remained with no modification, whereas new protocols below this layer have been designed to accommodate the communications between the M-ES and the MD-IS. The protocol stack below layer 3 at the M-ES is illustrated in Fig. 2.2, where the MAC layer protocol supports the connection between the M-ES and the MDBS, and the pair of protocols, including the Mobile Data Link Protocol (MDLP) and the Subnetwork Dependent Convergence Protocol (SNDCP), enables communications between the M-ES and MD-IS. The network layer supports both IP protocol and the OSI Connectionless Network Protocol (CLNP) to accomplish the routing of the IP/CLNP packets over the CDPD backbone.

In this thesis we mainly concentrate on the protocols in the three lower layers, including the physical (PHY) layer, the MAC layer and the LLC layer. The organization of this chapter is as follows. After this introduction, in Section 2.2 we will be describing the PHY layer configuration including the modulation/demodulation schemes and the communication channel. In Section 2.3 we will give detailed description of the MAC layer configuration, including how the link layer data packets are transformed into a bit stream at the MDBS, and how they are reconstructed at the M-ES. In this section we will also briefly address the medium access algorithm. Then in Section

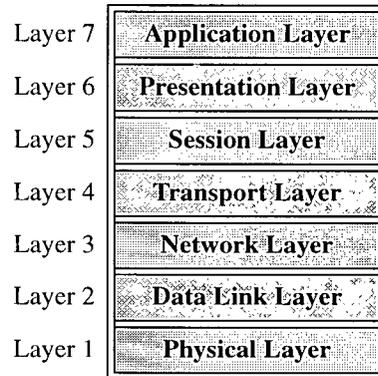


Figure 2.1 OSI reference model.

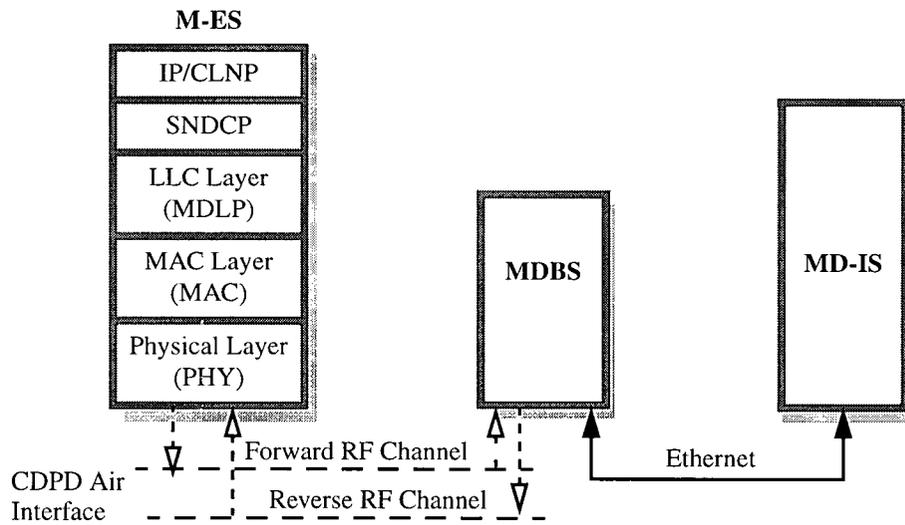


Figure 2.2 M-ES protocol stack.

2.4 the configuration of the logic link layer will be provided, together with the ARQ mechanism employed in this layer.

2.2 PHY Layer Configuration

As illustrated in Fig. 2.2, the PHY layer communication between an MDDBS and an M-ES takes place through the CDPD air interface. Furthermore, the communications between the MDDBS and the MD-IS takes place through wired connection, such as the Ethernet. In this section we mainly focus on the air-interface between the M-ES and the MDDBS. It consists of a pair of 30kHz-bandwidth RF channels, i.e. the forward channel and the reverse channel. The function of the PHY layer entities is to transform a sequence of bits from the MAC layer into a modulated waveform suitable for transmission onto the 30 kHz Radio Frequency (RF) channel, while being responsible for managing the radio resource and maintaining the signal power and communication quality. The raw data transmission rate that CDPD can provide on both forward and reverse channel is 19.2 kbps.

The physical layer modulation scheme employed on the CDPD RF channel is Gaussian Minimum-Shift Keying (GMSK), which is a popular member of the Continuous Phase Modulation (CPM) family with excellent spectral properties and simple implementation structures [16]. The following sections will describe the GMSK transmitter and receiver structures, as well as the communication model.

2.2.1 GMSK Transmitter

The block diagram of a GMSK transmitter is shown in Fig. 2.3. It consists of a Gaussian Low Pass Filter (GLPF) and a Frequency Modulator (FM). The input to the GLPF is a binary nonreturn to zero (NRZ) sequence². The GLPF is used to bandlimit the input pulses. The output

² A differential encoding process needs to be applied to the input signal before it is fed into the GLPF, if the signal is detected by a 2-bit or higher bit number differential detector [17]. In this thesis where the coherent detector and 1-bit conventional detector are employed, the presence of differential encoder is not required.

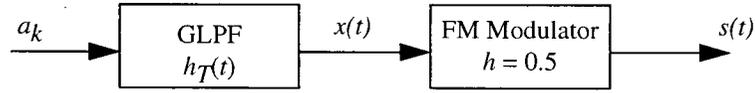


Figure 2.3 Block diagram of GMSK transmitter

signal $x(t)$ can be written as

$$x(t) = \sum_{k=-\infty}^{\infty} a_k g(t - kT) \quad (2.1)$$

where

$$g(t) = h_T(t) \otimes p(t) \quad (2.2)$$

where \otimes denotes a convolution, $p(t)$ is a rectangular pulse of duration T and unity amplitude, and $h_T(t)$ is the impulse response of the GLPF [19]. This impulse response can be mathematically expressed as:

$$h_T(t) = \frac{1}{\sqrt{\pi}} k_1 B_t \exp[-(k_1 B_t t)^2] \quad (2.3)$$

where $k_1 = \pi \sqrt{\frac{2}{\ln 2}} \approx 5.336$ and B_t is the 3-dB bandwidth of the GLPF. The output of the FM, the transmitted GMSK signal $s(t)$ can be mathematically expressed as

$$s(t) = A_o \cos[2\pi f_c t + \phi(t)] \quad (2.4)$$

where A_o is the constant amplitude, f_c is the carrier frequency and

$$\phi(t) = 2\pi h \sum_{k=-\infty}^{\infty} a_k \int_{-\infty}^t g(\tau - kT) d\tau \quad (2.5)$$

where h is the modulation index. For MSK-type signals we have $h = 1/2$ so that the maximum phase change over one symbol duration T is $\pi/2$ [20]. $g(t)$ is the input pulse response of the GLPF

which is normalized such that $\int_{-\infty}^{\infty} g(t) dt = \frac{1}{2}$. It is given by [18],

$$g(t) = \frac{1}{2T} \left\{ Q \left[k_2 B_f T \left(-\frac{1}{2} - \frac{t}{T} \right) \right] - Q \left[k_2 B_f T \left(\frac{1}{2} - \frac{t}{T} \right) \right] \right\} \quad (2.6)$$

where $k_2 = \sqrt{2}k_1 \approx 7.547$, $B_f T$ is the normalized (to symbol duration) 3-dB bandwidth of GLPF and $Q(\cdot)$ is the well known Q -function given by [21]

$$Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^{\infty} \exp\left(-\frac{\omega^2}{2}\right) d\omega. \quad (2.7)$$

The pulse shape and spectral properties of GMSK signals heavily depend on the $B_f T$ product. Fig. 2.4 illustrates the plots of $g(t)$ and corresponding power spectrum with $B_f T$ as a parameter. Different values of the $B_f T$ product have been chosen for different applications. For example, GSM has chosen $B_f T = 0.3$ as the standard, while in the CDPD standard that we are investigating in this thesis, $B_f T = 0.5$ has been adopted [1].

2.2.2 Communication Channel Model

In urban areas, the mobile radio environment between an M-ES and an MDBS can be characterized as a multipath medium, where the transmitted signal waves are reflected from

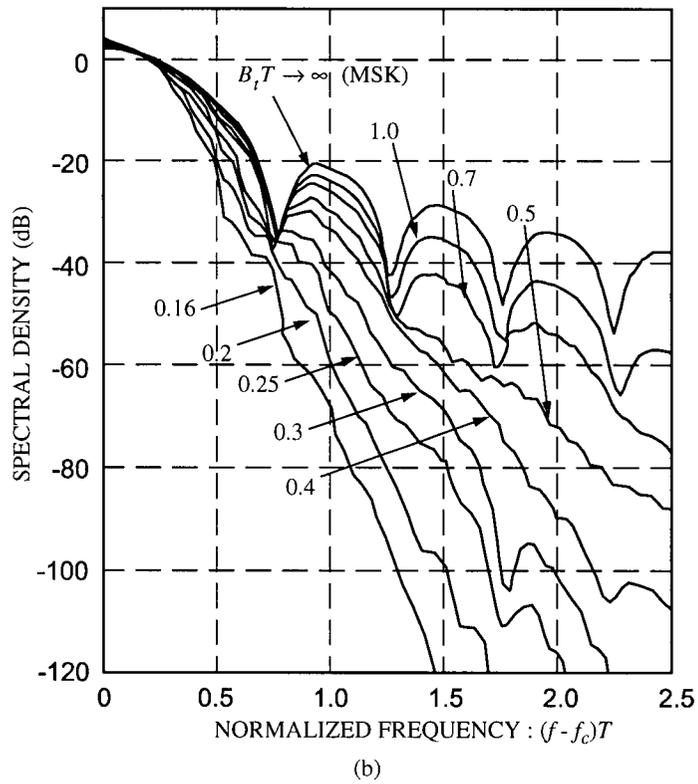
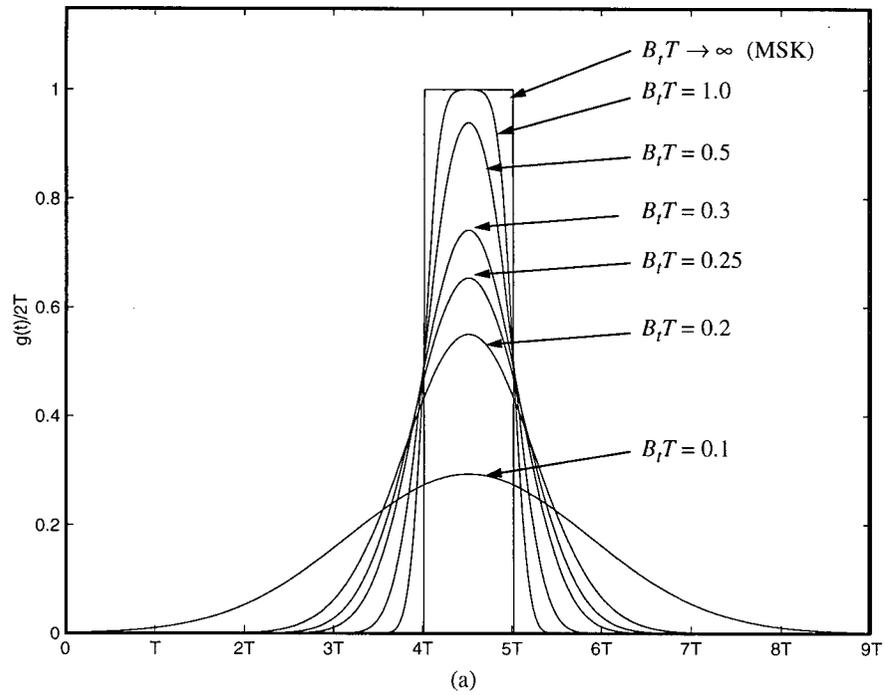


Figure 2.4 (a) Pulse response of Gaussian lowpass filter. (b) Corresponding power spectral of GMSK signals [16]

nearby objects or buildings [23]. Therefore, in most cases, the M-ES receives no signal from the direct path but a sum of scattered signals with various strengths and delays, resulting signal amplitude fluctuation and phase randomization. This dispersive medium is a typical land-mobile channel, referred to as multipath fading or Rayleigh fading channel, where the amplitude r of the faded carrier is a Rayleigh distributed random process and the signal phase θ is uniformly distributed from 0 to 2π [22]. Mathematically this can be expressed as,

$$f(r) = \begin{cases} \frac{r}{b} \exp\left(-\frac{r^2}{2b}\right), & \text{when } r \geq 0 \\ 0, & \text{when } r \leq 0 \end{cases} \quad (2.8)$$

$$f(\theta) = \frac{1}{2\pi}$$

where $f(\cdot)$ denotes the probability density function (pdf), and b is the mean power [22]. A fading channel is associated with a multipath spread which represents the time delay spread of the incoming multipath signals. If the multipath spread relative to the symbol duration is small enough so as not to cause any Inter-Symbol Interference (ISI), the fading channel can be classified as frequency-nonselctive or flat fading, and as frequency-selective fading if vice versa [24]. Since a CDPD radio channel is 30-kHz narrow band, we will only consider flat fading in our work. Furthermore, the M-ES in motion also suffers from the Doppler frequency shift, which spreads the received carrier frequency. The maximum Doppler shift frequency is determined by the velocity V of the mobile unit, the transmitted carrier frequency f_c , and the speed of light $c \approx 3 \times 10^8$ m/s, i.e.

$$f_D = \frac{V f_c}{c} . \quad (2.9)$$

Let us consider the value of the normalized Doppler shift bandwidth $f_D T$. Suppose the mobile is moving in 100 km/hour, assume the carrier frequency is 800 MHz, and the transmission data rate is 19.2 kbps, then the resulting $f_D T$ is approximately 0.0039, which is a quite small value. Since the mobile's velocity is normally less than 200 km/hour, according to the results illustrated in this example, we can consider that our system is operating in a slow fading environment.

A block diagram of the used channel model is shown in Fig. 2.5. The transmitted signal $s(t)$ is multiplied by the fading signal $f(t)$, the resulting signal is then corrupted by the AWGN $n(t)$ with double-sided power spectral density of $N_0/2$. The fading signal is generated by the well-known Jakes' fading simulator [22].

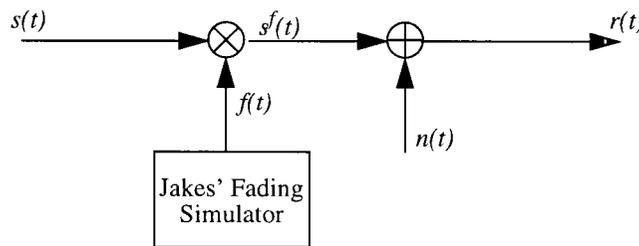


Figure 2.5 Block diagram of communication channel model

2.2.3 GMSK Receivers

GMSK receivers can be classified into two categories, i.e. coherent receivers [16] and noncoherent receivers, which can be either differential receivers or limiter/discriminator receivers [17]. Coherent detection exploits knowledge of the carrier's phase reference and thereby providing the optimum error performance over AWGN channel. However, in real mobile communica-

tion channels, non-coherent detection such as differential detection, which does not require carrier recovery, exhibits a better error performance with the presence of channel interference including fading. Beyond conventional differential detectors, there are also other types of implementations with various enhancements depending on particular performance requirements, such as the 2-bit differential detector with/without decision feedback [25]. However, for the purpose of this thesis we will consider only the performance with both a conventional coherent receiver and a conventional 1-bit differential detection receiver.

Coherent receiver: The block diagram of the coherent receiver is depicted in Fig. 2.6 [16]. The signal is passed through the post-detection filter, which is a Gaussian bandpass filter (GBPF) with $B_r T = 0.63$ [16]. The band-limited signal is then quadrature demodulated into the in-phase and quadrature-phase components $I(t)$ and $Q(t)$. The I, Q baseband signals are alternatively sampled at $2T$ intervals. Finally the sampled data sequence are fed into a bit decision logic to generate the final information bits.

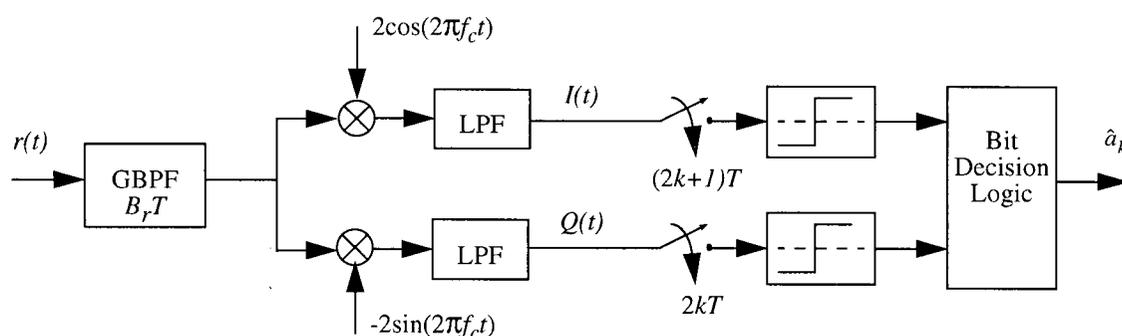


Figure 2.6 Block diagram of coherent GMSK detector

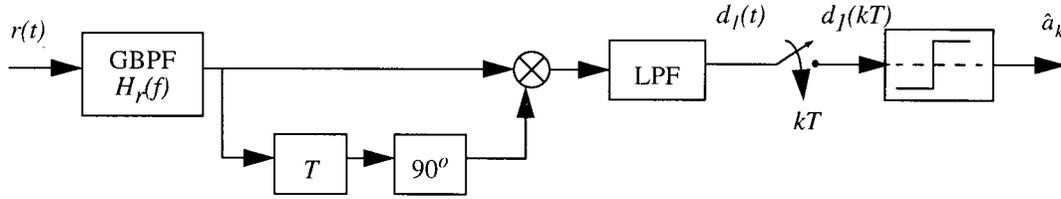


Figure 2.7 Block diagram of GMSK 1-bit conventional differential detector

Differential receiver: The block diagram of the conventional 1-bit differential detector is depicted in Fig. 2.7 [26]. From the channel model, the received signal can be expressed as the sum of the Rayleigh-faded signal $s^f(t)$ and the AWGN $n(t)$ as,

$$r(t) = s^f(t) + n(t). \quad (2.10)$$

The signal-to-noise ratio (SNR) can be expressed as [18]

$$SNR = \frac{A_o^2}{2\sigma_n^2} = \frac{E_b}{N_o} \cdot \frac{a^2(t)}{B_{rn}T} \quad (2.11)$$

where $B_{rn} = 1.026B_r$ is the normalized noise bandwidth of the predetection receive filter $H_r(f)$, which is a 4-th order bandpass Butterworth filter with $B_rT = 1.0$ [26], and $a(t)$ is the normalized signal amplitude after the signal is filtered [18], i.e.

$$a(t) = \sqrt{[h_r(t) \otimes \cos(\phi(t))]^2 + [h_r(t) \otimes \sin(\phi(t))]^2}. \quad (2.12)$$

As illustrated in Fig. 2.7, the bandpassed signal is multiplied by its own version which is one-bit delayed and $\pi/2$ -phase shifted. The sampler output is obtained by lowpass-filtering and sampling the product of the one-bit detector. If the effects of fading, noise, and intersymbol

interference due to filtering are ignored, the normalized sampled value $d_1(kT)$ can be expressed as [26],

$$d_1(kT) = \sin(\Delta\phi_k) \quad (2.13)$$

where

$$\Delta\phi_k = \sum_{i=-\infty}^{\infty} b_i\phi_{k-i} \quad (2.14)$$

is the differential phase angle, with which the decision can be made by

$$\hat{a}_k = \text{sgn}[d_1(kT)] \quad (2.15)$$

where \hat{a}_k is the estimate of information bits a_k , $\text{sgn}[x] = 1$ for $x \geq 0$ and $\text{sgn}[x] = -1$ for $x < 0$.

2.3 MAC Layer Configuration

The MAC layer entities logically operate between the PHY and LLC layers to convey information, namely, link protocol data units (LPDUs), between peer LLC entities across the CDPD air interface. It provides the following primary services [3]:

- Encapsulates the LPDUs into frame structures to ensure LPDU delimiting, frame synchronization and data transparency;
- Encodes the sequence of frames to provide error protection against mobile channel impairments;
- Detects and corrects bit errors within received frames;

- Arbitrates access to the shared reverse channel;
- Synchronizes with the forward channel transmissions to make feasible the reception of data as well as control information transmitted in every CDPD cell.

In Chapter 1 we have discussed several main causes of performance degradations in CDPD airlink, and argued that it is important and necessary to study the performance of the forward channel MAC layer. Before we start, it is instructive to outline the MAC layer configuration and its primary functions in the forward channel. In the following subsection we will describe in detail how the LPDUs are transformed into a continuous data sequence, namely, the Forward Channel Transmission Sequence (FCTS) at the MDBS. Then we will describe the procedure of recovering the received LPDUs at the M-ES in Section 2.3.2. Finally in Section 2.3.3 we will briefly address the medium access algorithm which is employed at the MAC layer to arbitrate access to the shared reverse channel.

2.3.1 MAC Layer Configuration at the MDBS

Let us consider a number of LPDUs that need to be transmitted by the MDBS. As illustrated in Fig. 2.8, these LPDUs are first flag-delimited using the well-known flag pattern 01111110 at the MAC layer [2]. Subsequently, zero-stuffing takes place to ensure data transparency and then the LPDUs are linked together to form a continuous frame data bit stream. It should be noted that the transmission of this bit stream is never interrupted. For example, even when there are no LPDU data for transmission, instead either control LPDUs or sequences of contiguous flags are transmitted. Hence, a contiguous sequence of data packets is continuously transmitted on the forward channel, and is referred to as the Continuous Frame Data Bit Stream (CFDBS).

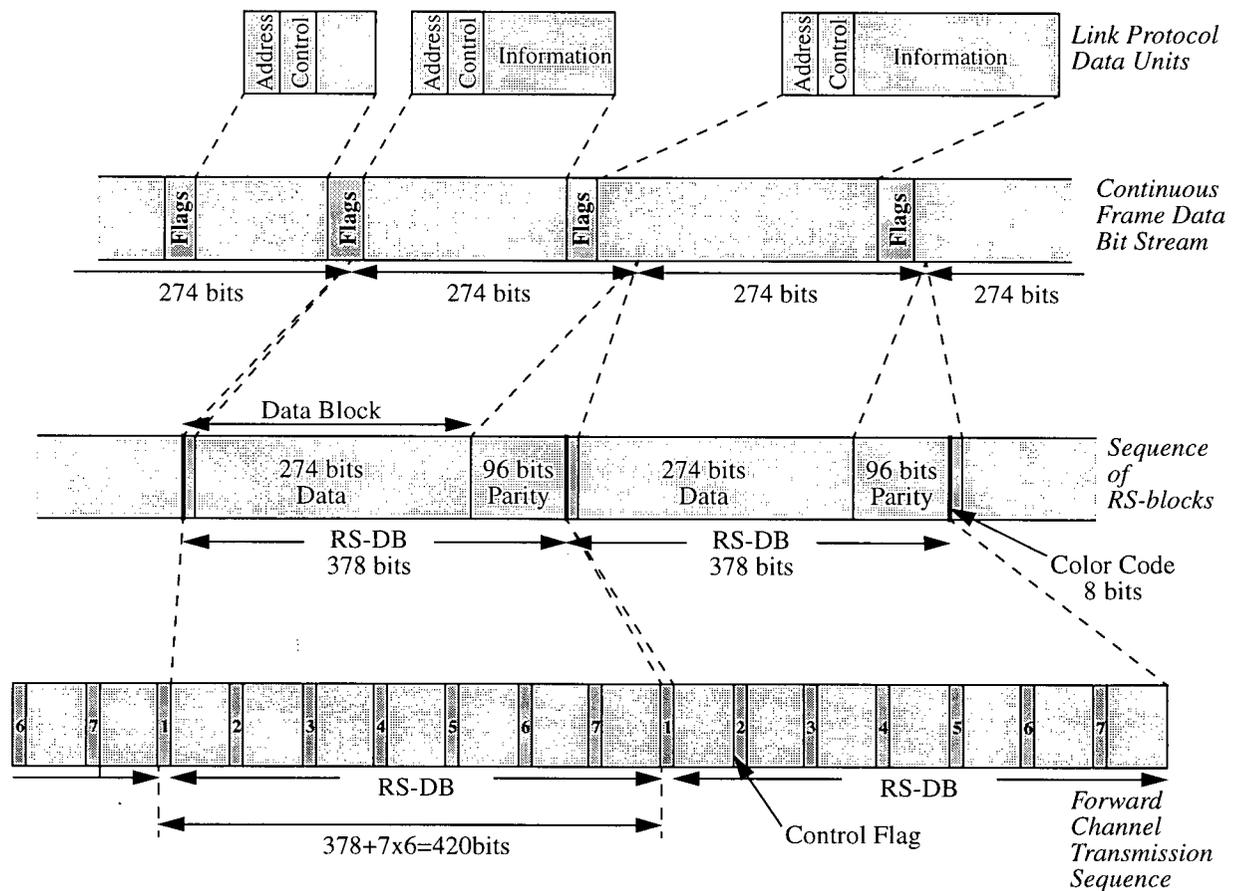


Figure 2.8 Data transmission procedures in the forward channel.

Following the CDPD standard, the CFDBS is divided into segments of 274 consecutive bits and each segment is prefixed by an 8-bit Color Code. In this way, a series of consecutive data blocks is formed each one with a fixed length of 282 bits. The Color Code is a special pattern assigned to every individual CDPD channel stream and is used for co-channel interference detection. For each cell, all RF channels available for CDPD are assigned the same value of Color Code [2]. The data blocks are then encoded using a systematic (63,47) Reed-Solomon (RS) error correcting code. From the encoding point of view, each data block represents an information field of 47 6-bit symbols (or codewords). The encoding of this information field generates a 16-symbol

parity field (96 bits), which is appended at the end of each data block. Thus, a consecutive sequence of RS-encoded data blocks (RS-DB) is generated, as shown in Fig. 2.8. These RS-DBs, each one with a fixed length of 378 bits, form the basic transmission units of the forward channel. Note that this (63,47) RS coding is common to both the forward channel and the reverse channel and is capable of correcting up to 8 erroneous symbols within one block [7]. Prior to actual transmission on the forward channel, each RS block is passed through a 9th order scrambler with a generator polynomial, $g(x) = x^9 + x^8 + x^5 + x^4 + 1$. This process reduces the likelihood to have long strings of binary ones and zeroes within the transmission bit stream. Such long strings are generally avoided because they are not easily tracked by certain types of demodulators (e.g. phased locked loops) and may result in reduced performance or increased implementation complexity.

A general flow chart describing the procedure of generating the Forward Channel Transmission Sequence (FCTS) is illustrated in Fig. 2.9a. It should be noted that the actual FCTS is the contiguous sequence of RS-DBs (after scrambling) interleaved with special Control Flags. These flags carry synchronization information that helps M-ESs acquire block synchronization and decode the forward channel, as well as MAC-level control information that helps M-ESs effectively share the common reverse channel.

2.3.2 Packet Reconstruction at the M-ES

The flow chart of the procedure of LPDU reconstruction at the M-ES is shown in Fig. 2.9b. The received bit sequence is formed at the M-ES after signal demodulation and detection at the PHY layer. After control flag extraction and descrambling, which are the reverse processes to recover the RS-DBs at the MAC layer, the RS-DBs are Reed-Solomon decoded. The operation

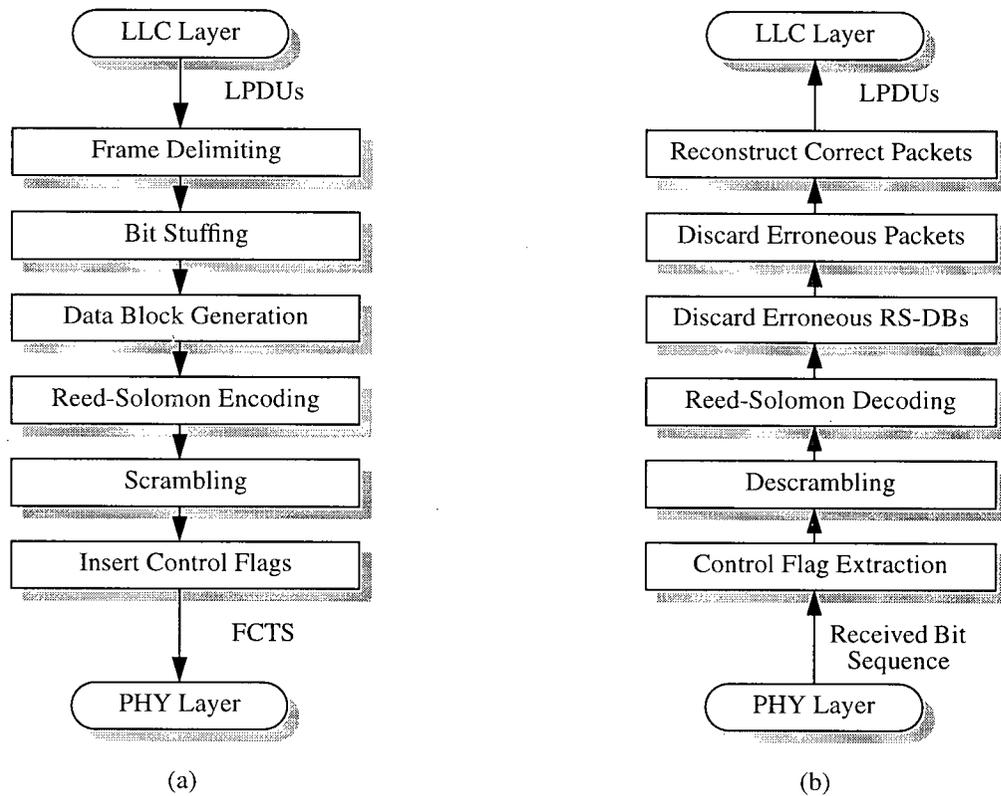


Figure 2.9 Flow chart for (a) procedure of FCTS generation at the MDBS; (b) procedure of LPDU reconstruction at the M-ES

involves error detection and correction. Those RS-DBs that contains too many errors, i.e. more than 8 codewords in a block are in error, are identified and discarded at the output of the decoding process. By identifying the flags delimiting the LPDUs, the erroneous LPDUs (packets) can be identified from the sequence of the correct data blocks. Since the MAC layer is not responsible for maintaining the quality of service, i.e. it is not responsible for erroneous packet retransmission, those erroneous packets will be discarded. Only the correct ones are kept for final processing, including the delimiting flag extraction and de-zero-stuffing. The resulting LPDUs are then passed to the LLC layer for further processing.

2.3.3 Medium Access Algorithm

The reverse channel is shared by a number of M-ESs within a cell. In order to avoid collisions, the M-ES who wants to establish connection to the MDBS has to follow a certain type of rule defined by an medium access algorithm. The algorithm employed in the CDPD system is called Digital Sense Multiple Access with Collision Detection (DSMA/CD), which regulates the procedure of M-ESs accessing the reverse channel. This algorithm is similar to the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) used in IEEE 802.3 LANs such as Ethernet. However the M-ESs cannot sense the reverse channel directly, because the reverse channel in CDPD employs different reception and transmission frequency band. In the DSMA/CD the M-ES senses the Busy/Idle flag which is transmitted periodically in the forward channel. When the reverse channel is idle, it starts transmission until the Block Decode Status Flag transmitted in the forward channel signals the M-ES a decoding failure, which is caused either by collision or channel burst errors. Then the transmitting M-ES (or M-ESs) attempts to regain access to the channel after an appropriately selected exponential backoff delay.

2.4 Logic Link Layer and ARQ Procedure

In a point-to-point information transfer, the CDPD can provide reliable communication through acknowledged operation, with which the network layer information is transmitted in frames that are acknowledged at the LLC layer. This layer employs the MDLP protocol, which utilizes the services of the MAC layer to provide access to the physical channel and transparent transfer of link-layer frames between data link layer entities. The MDLP implemented in an M-ES communicates with a peer MDLP located in its serving MD-IS. The MDBS functions primarily as a data link relay between an MD-IS and M-ESs except in some cases where the MDBS may also operate in the LLC layer [1].

To ensure reliable communication between MDLP peers, an ARQ mechanism is employed as part of the MDLP protocol to perform tasks such as identifying missing frames and sending requests for their retransmission. The details of the ARQ procedure will be discussed in Section 2.4.2, following the Section 2.4.1 where we will provide the descriptions of the frame structure in the LLC layer.

2.4.1 LLC Layer Frame Structure

Information transfer between peer LLC entities is carried out through a number of Link Protocol Data Units (LPDUs). A general structure of the LPDUs is given in Fig. 2.10. The frame consists of three fields, namely, the Address Field, Control Field, and the Information Field. The Address Field is 1-4 octet long, which contains information to specify the virtual data link channel. The address information includes a Temporary Equipment Identifier (TEI). By assigning different values to the TEI, the data link connection can be specified as either point-to-point or broadcast. For point-to-point connection, the TEI is used as the M-ES data link address, which is typically assigned to each M-ES uniquely.

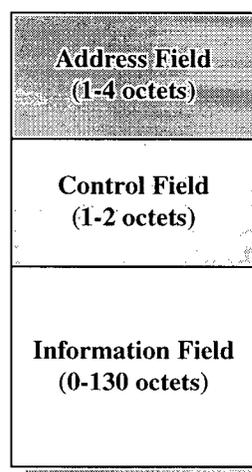


Figure 2.10 Structure of the LLC frames

The Control Field is 1-2 octets long. It contains the frame type identifier as well as sequence numbers where applicable. There are three generic types of frames: numbered information (I) frames, supervisory (S) frames and unnumbered (U) frames. The I-frames are used to carry acknowledged information transfer. The S-frames are used to perform data link supervisory control functions such as acknowledge I-frames, or request retransmission of I-frames. The U-frames are used to provide additional control functions and to carry unacknowledged information transfer. The information field is optional. Some types of frames, for example, S-frames, do not require information field. The size of this field varies from 0 octet to 130 octets [1].

2.4.2 ARQ Procedure

In acknowledged transmission mode, The MDLP is responsible for identifying and requesting retransmission of the missing data frames. This operation is referred to as the ARQ procedures. At the receiving side, the LPDUs are reconstructed with the RS-DBs decoded at the MAC layer. These frames containing uncorrectable errors are discarded while the recovered ones are forwarded to the LLC layer, where the MDLP identifies the missing frames by checking their sequence numbers carried in the control field of the LPDUs. In the case of receiving an out-of-sequence I-frame, or the expiry of a retransmission timer, the receiving side will send an S- frame to its peer LLC entity requesting for retransmission.

The information transfer on the data link is based on a sliding-window (with a window size of 128 frames) protocol with a Selective-Reject (S-REJ) ARQ scheme for error recovery. With S-REJ ARQ, the receiving side sends a negative acknowledgment (NAK) for each missing I-frame. The only frames retransmitted are those that receive a NAK or which time out. It is more efficient than other conventional ARQ approaches such as stop-and-wait ARQ and go-back-N

ARQ, since it minimizes the amount of retransmission [29].

2.5 Computer Simulation Model and Parameters

A general block diagram of our computer simulation system considered in this thesis is given in Fig. 2.11. It consists of three sub-systems, namely, one MDBS, the communication channel, and one M-ES. The MDBS continuously transmits a sequence of data frames through the forward mobile communication channel. The M-ES receives these frames and tries to decode them. Because the system performance study is focused on the forward channel, the reverse channel is assumed ideal, such that there is no transmission failure on the reverse channel and the transmission time is ignored.

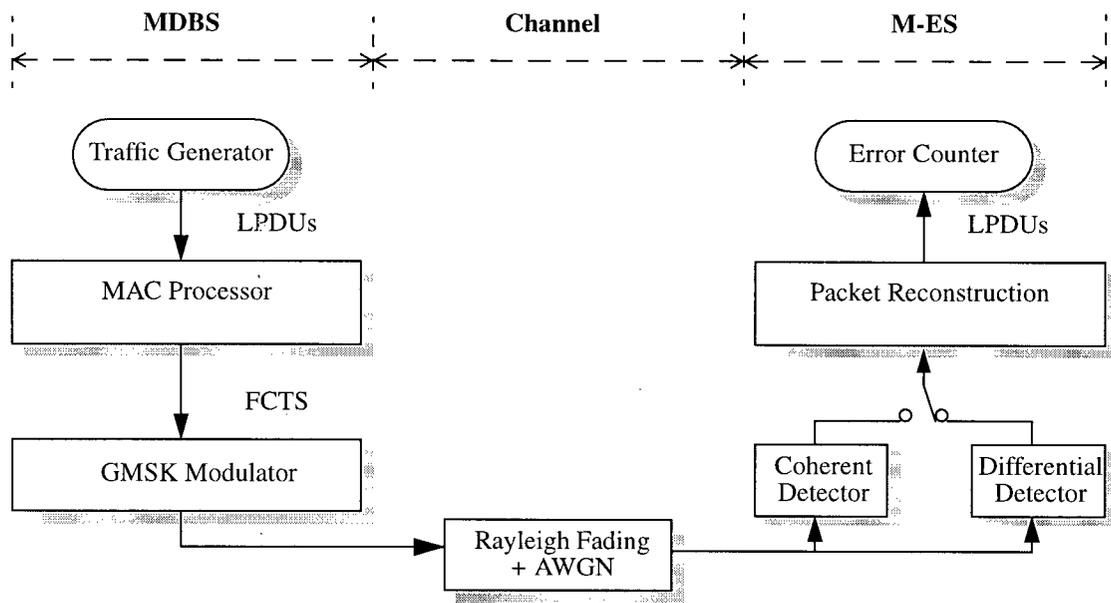


Figure 2.11 Block diagram of computer simulation model

2.5.1 MDBS Model

The traffic generator at the MDBS generates a number of LPDUs, which are transformed

into the FCTS by the MAC processor. The function of the MAC processor has been detailed in Section 2.3.1. Then the transmission sequence is modulated by a GMSK modulator with the normalized pre-modulation filter bandwidth $B_p T = 0.5$.

Since in practice the lengths of the LPDUs generated from the LLC layer are dynamically dependent on the types of the frames and the lengths of the message source from the upper layer, the frame lengths can be considered as random. For simplicity purposes, we choose the following three typical lengths, which will be randomly picked by the traffic generator:

- (1) Small Packet (SP) with length of 6 octets or 48 bits, corresponding to the length of a supervisory frame, e.g. an acknowledgment frame;
- (2) Medium Packet (MP) with length of 71 octets or 568 bits, corresponding to the average frame length, i.e. the mean value of the lengths of a Small Packet and a Large Packet;
- (3) Large Packet (LP) with length of 136 octets or 1088 bits, corresponding to the default data frame length [1], which is also the maximum length allowed for an LPDU.

For the sake of simplicity, the three kinds of frames are randomly chosen with equal probability, i.e. each with probability of 1/3.

2.5.2 Communication Channel Model

The modulated GMSK signal is passed through the communication channel which we have considered in this thesis to be a narrowband land-mobile Rayleigh fading channel with a maximum Doppler shift f_D . The statistical properties of the channel have been detailed in Section 2.2.2. In this thesis where slow fading is considered, the following three values of the maximum

Doppler shift f_D have been chosen: 20Hz, 40Hz and 80Hz. Since the bit rate of a CDPD air link is 19.2 kbps, these values correspond to the normalized Doppler shift bandwidth $f_D T$ of approximately 0.001, 0.002, and 0.004, respectively. The three values of f_D represent the typical mobile moving speeds, e.g. assuming a carrier frequency of 800 MHz, the three Doppler frequencies correspond to 25, 50, and 100 km/hour respectively. Also, as previously mentioned, besides Rayleigh fading, the transmitted signal is also corrupted by the AWGN, with a 2-sided power spectral density of $N_0/2$.

2.5.3 M-ES Model

As it shown in Fig. 2.11, the channel-corrupted signal is detected at the mobile station by either a differential detector or a coherent detector. As previously mentioned, we will consider both types of detectors into account in order to illustrate the differences in terms of their performances under various channel conditions. The detected bit sequence is then passed through a number of processes that perform reconstruction of the correctly received packets. The reconstruction operations have been depicted in Section 2.3.2. The function of the Error Counter is to not only count the number of erroneous bits, RS-DBs, and LPDUs, but also to measure the amount of information data successfully received at the M-ES. The detailed performance considerations will be presented in the next chapter.

2.6 Summary

In this chapter the CDPD system model has been described. Descriptions of lower layer protocol sets, including MDLP which operates at the LLC layer, and MAC protocol which operates on the medium access layer, have been provided. The physical layer configuration, including GMSK modulation/demodulation scheme, and the mobile radio channel model, has

also been addressed. The functions of the CDPD communication units, such as MDBS, MD-IS, and M-ES are given, and how the transmission sequence is formed in the forward channel are detailed in this chapter.

Chapter 3 Performance Analysis and Evaluation

3.1 Introduction

Given the system model presented in the previous chapter, the main objective of this thesis is to study the performance of the pre-described CDPD system in terms of the BLER, the PER, and the MAC layer throughput, over various channel conditions and receiver types. Hence, in this chapter we will investigate the performance of BLER and PER and their relations. We also identify a source of performance degradation, namely, the correct but unusable data (CUD), which is caused by the variety of the packet lengths and the alignment of the packets and RS-blocks at the MAC layer.

The organization of this chapter is as follows. After this introduction, we will describe the source of degradation, i.e. the CUD in Section 3.2. Then the relations between the BLER and PER will be analyzed in Section 3.3, followed by the derivation of the throughput expression in Section 3.4. Numerical results of computer simulations and discussions will be presented in Section 3.5. Finally, a summary of the chapter appears in Section 3.6.

3.2 Correct but Unusable Data

One of the major problems addressed in this thesis is the fact that if one RS block is destroyed, i.e. it is not recovered by the MAC error correction scheme because it entails a large number of errors which cannot be corrected by the available coding scheme, it causes one or more packets to be discarded as unusable. As an example, let us consider the scenario depicted in Fig. 3.1, the destroyed RS block, B4, happens to contain data from two packets, P3 and P4. Both these packets will be discarded at the MAC layer even though only a small fraction of them is unrecov-

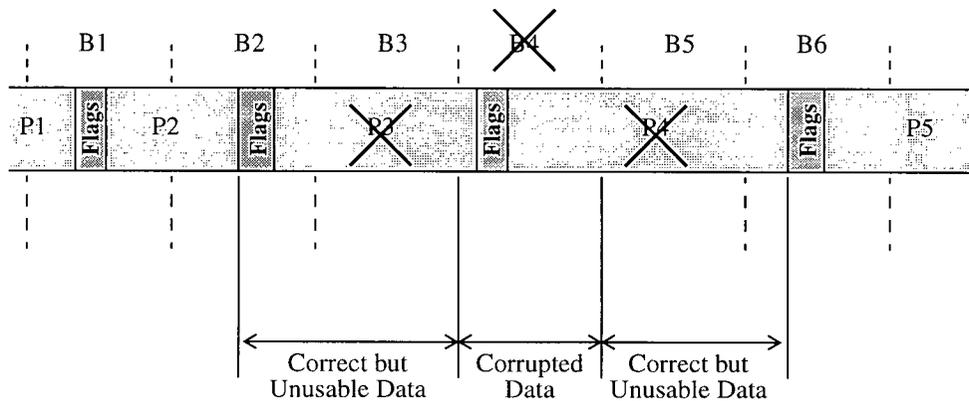


Figure 3.1 Generation of correct but unusable data (CUD)

× : Destroyed block

erable. This means that the receiver discards data from additional RS blocks that are not corrupted. In a reliable transmission link with ARQ employed, the discarded packets containing the portion of data that is correct but unusable would have to be retransmitted. Evidently, this fact represents a source of inefficiency.

In general, we may argue that a percentage of data arriving at the receiver without being destroyed is discarded as unusable. We refer to this data as correct but unusable data (CUD). To demonstrate the effect of CUD, we define the percentage of CUD as

$$\eta = 100\% \cdot \frac{1}{N} \sum_{k=1}^N e_k \quad (3.1)$$

where N is the total number of transmitted bits, $e_k = 1$ if the k -th bit belongs to the CUD data, and $e_k = 0$ otherwise. The relations between the percentage η and the energy per bit-to-noise ratio E_b/N_o can be illustrated in Fig. 3.2. This graph is derived through computer simulation using the

model depicted in Fig. 2.11, where a sequence of packets are transmitted through the AWGN channel and received by a coherent detector. As expected, CUD is very low, e.g. less than 10%, at very small and very high values of E_b/N_o . During very small values, almost all received RS-DBs contain a large number of errors and cannot be recovered; hence undestroyed data is close to zero. On the other hand, during high values, almost all RS-DBs are correctly received and no blocks are discarded, hence no useless data exist. However, it is interesting to note that, even in the AWGN channel, the percentage of CUD approximates a peak between 40% and 45% when $E_b/N_o \cong 3.5$ dB. Equivalently, this means that nearly 45% of the received data is discarded by the receiver although this data is not destroyed. Clearly, this represents a significant redundancy and this is a source of performance degradation.

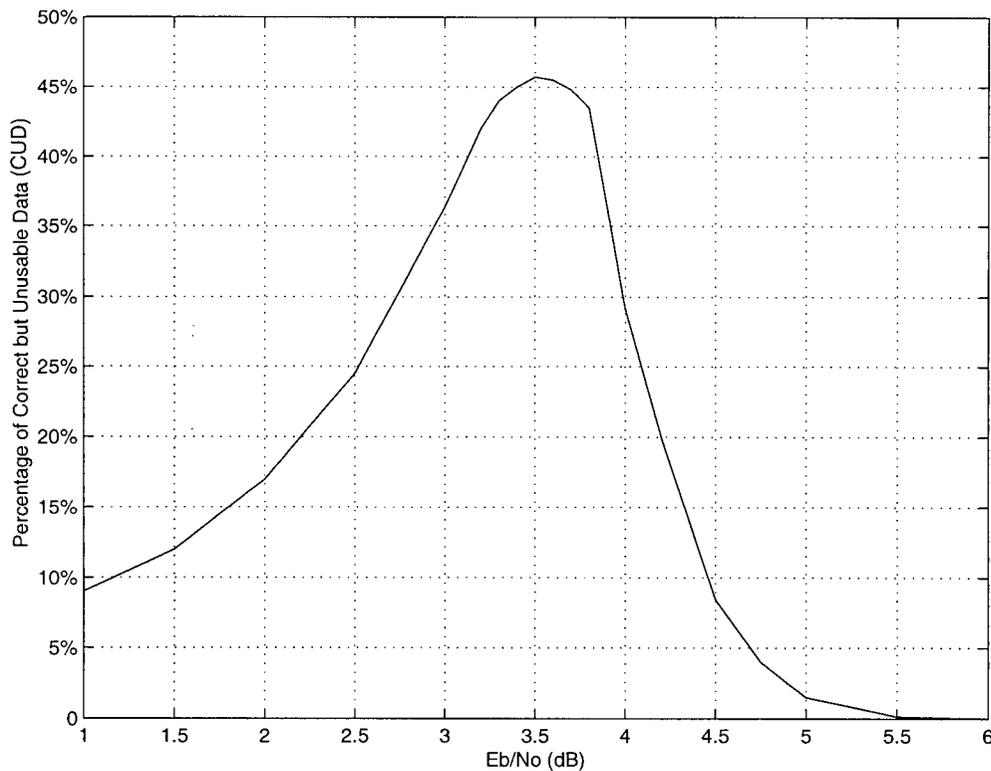


Figure 3.2 Percentage of CUD in the AWGN channel with coherent detection

As will be presented in Section 3.5.2, simulation results of the percentages for the Rayleigh fading channel have also demonstrated similar inefficiency by 40% to 45% of the CUD. To effectively improve the performance of CDPD, it is mandatory to mitigate or even eliminate the above problem. In doing so, we have proposed a new scheme that effectively overcomes the aforementioned deficiencies, by means of an ARQ protocol that operates at the MAC layer. A detailed presentation and analysis of this protocol will be given in Chapter 4.

3.3 Block and Packet Error Performance

In this section we address the relation between the BLER and the PER. It is instructive to consider this relationship in order to study how the packet error process is affected by the underlying block error process. A simple theoretical performance analysis with an AWGN channel is considered first.

Let us consider the probability a packet to be destroyed, i.e. the probability one or more blocks containing data from that packet to entail a large number of errors and be unrecoverable at the receiver. Let P_P denote the probability one data packet to be destroyed and let P_B denote the probability one block to be destroyed. Given our simulation parameters, i.e. the length of a SP is 48 bits, the length of an MP is 568 bits, and the length of an LP is 1088 bits, we have: i) A Small Packet can occupy either 1 or 2 blocks, ii) A Medium Packet can occupy either 3 or 4 blocks, and iii) A Large Packet can occupy either 4 or 5 blocks. Therefore, the probability of one SP to be destroyed is given by:

$$P_S = \text{Prob}\{X_{e,SP}|X_{o,SP,1}\} \cdot \text{Prob}\{X_{o,SP,1}\} + \text{Prob}\{X_{e,SP}|X_{o,SP,2}\} \cdot \text{Prob}\{X_{o,SP,2}\} \quad (3.2)$$

where $X_{e,SP}$ represents the event that an SP is destroyed, and $X_{o,SP,n}$ represents the event that an

SP occupies n block(s), $n = 1, 2, 3, \dots$. Considering that for a memoryless channel, such as the AWGN channel,

$$\text{Prob}\{X_{e,SP}|X_{o,SP,n}\} = 1 - (1 - P_B)^n \quad n = 1, 2, 3, \dots \quad (3.3)$$

where P_B is the probability a block is destroyed, and given the 274 bits packet length, Eq. (3.2)

can be rewritten as

$$P_S = P_B \cdot \frac{227}{274} + [1 - (1 - P_B)^2] \cdot \frac{47}{274}. \quad (3.4)$$

Likewise, the equivalent probabilities for a Medium Packet, P_M , and a Large Packet, P_L , can be respectively written as:

$$\begin{aligned} P_M &= \text{Prob}\{X_{e,MP}|X_{o,MP,3}\} \cdot \text{Prob}\{X_{o,MP,3}\} + \\ &\quad \text{Prob}\{X_{e,MP}|X_{o,MP,4}\} \cdot \text{Prob}\{X_{o,MP,4}\} \\ &= [1 - (1 - P_B)^3] \cdot \frac{255}{274} + [1 - (1 - P_B)^4] \cdot \frac{19}{274} \end{aligned} \quad (3.5)$$

and

$$\begin{aligned} P_L &= \text{Prob}\{X_{e,LP}|X_{o,LP,4}\} \cdot \text{Prob}\{X_{o,LP,4}\} + \\ &\quad \text{Prob}\{X_{e,LP}|X_{o,LP,5}\} \cdot \text{Prob}\{X_{o,LP,5}\} \\ &= [1 - (1 - P_B)^4] \cdot \frac{9}{274} + [1 - (1 - P_B)^5] \cdot \frac{265}{274} \end{aligned} \quad (3.6)$$

Thus, the probability a packet to be destroyed is given by,

$$P_P = \frac{1}{3}P_S + \frac{1}{3}P_M + \frac{1}{3}P_L. \quad (3.7)$$

Substituting Eqs. (3.4)-(3.6) into Eq. (3.7), yields the expression of P_P as a function of P_B

$$P_P = \frac{1}{822}(2523P_B - 3630P_B^2 + 3017P_B^3 - 1353P_B^4 + 265P_B^5). \quad (3.8)$$

The curve corresponding to Eq. (3.8) is illustrated in Fig. 3.3 (solid line), which is verified by the simulation results shown in circle points. For comparison purposes, we have included the straight dotted line corresponding to the ideal situation where P_P and P_B are equal (i.e. each block contains only one packet) and therefore a block loss introduces only one packet loss, which is the minimum. It is interesting to note that the probability a packet to be destroyed, P_P , increases rapidly as P_B increases. For example, note that when P_B is about 0.1, P_P is almost 0.3, which is considered unacceptable in nearly every operating scenario. In addition, P_P is always larger than P_B . This is because a block loss always causes one or more packet loss. Therefore, a strong block correction mechanism is necessary in order to keep P_P quite small so that for satisfactory performance for the forward channel can be achieved.

Let us now study P_P and P_B with respect to the communications channel quality, typically measured as a function of E_b/N_o . It is well-known that an (N,K) Reed-Solomon block is encoded with N codewords including K codewords of information part and $(N-K)$ codewords of parity-check part [33]. Each codeword is l -bit long. For the AWGN channel the probability of an l -bit codeword in error is

$$P_{ecw} = 1 - (1 - P_e)^l. \quad (3.9)$$

An ideal RS code is capable of correcting up to $t = (N-K)/2$ codewords, therefore, the probability of an RS block is destroyed can be written as [35]

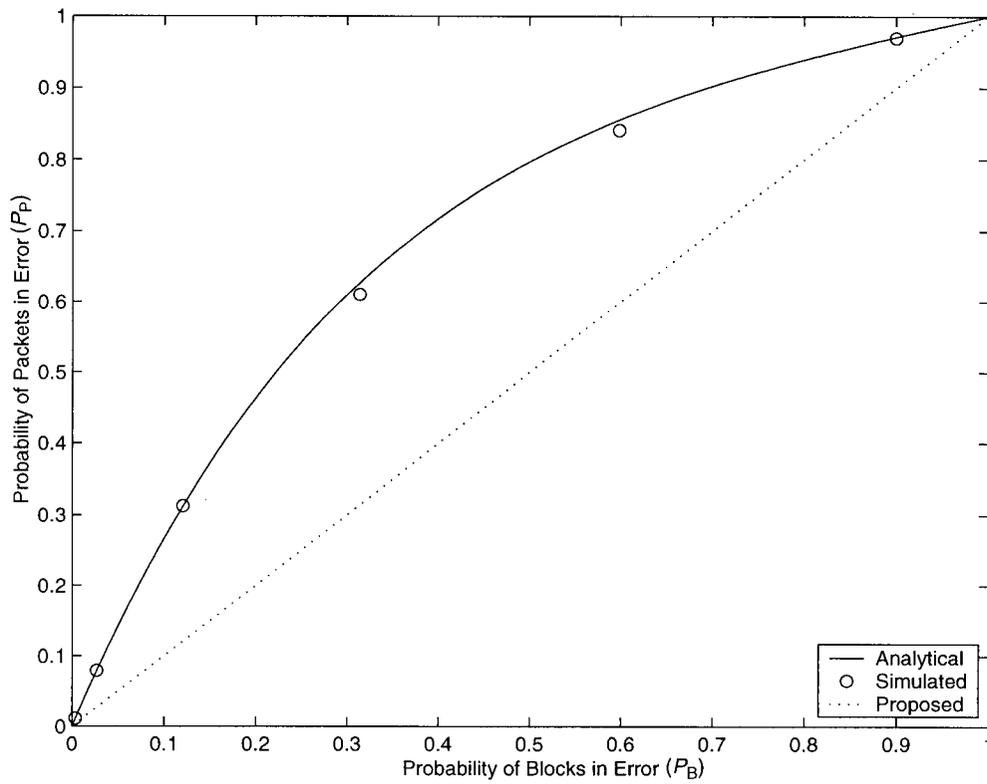


Figure 3.3 Probability of a packet in error (P_p) versus Probability of a block in error (P_B)

$$P_B = 1 - \sum_{k=0}^t C_k^N P_{ecw}^k (1 - P_{ecw})^{N-k} \quad (3.10)$$

where

$$C_k^N = \frac{N!}{(N-k)!k!} \quad (3.11)$$

Substituting Eq. (3.9) into Eq. (3.10), P_B can be expressed with the bit error probability P_e

as

$$P_B = 1 - \sum_{k=0}^t C_k^N [1 - (1 - P_e)^l]^k (1 - P_e)^{l(N-k)} \quad (3.12)$$

From [16], the bit error probability of GMSK with coherent detection can be approximated as a function of γ , which denotes the E_b/N_o , in the AWGN environment.

$$P_e(\gamma) \cong \frac{1}{2} \operatorname{erfc}(\sqrt{\alpha\gamma}) \quad (3.13)$$

where $\operatorname{erfc}(\cdot)$ is the complementary error function given by

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} \exp(-t^2) dt \quad (3.14)$$

and α is a constant parameter varied with different B_fT products. α can be estimated as 0.84 according to our observation of the simulation results over various values of α . Therefore, substituting Eq. (3.13) into Eq. (3.12) yields the expression of the P_B in terms of γ . And the expression of the packet error probability P_P in terms of γ can be easily obtained by substituting the resulting expression into Eq. (3.8).

3.4 Throughput Performance

In order to study the performance of the forward channel with demands to accommodate heavy data load, we focus our investigation on the throughput of the information data rather than the control overhead at the MAC layer. We will define the throughput as the average number of information bits transmitted successfully during one bit time. In other words, it can be calculated by dividing the total number of information bits to be transmitted, N_I , by the total number of bits actually transmitted over the channel N_{Total} , i.e.

$$S_{MAC} = \frac{N_I}{N_{Total}}. \quad (3.15)$$

In order to derive a simple expression, let us assume the packets have equal length L_P , then $N_I = n(L_P - H_P)$, where H_P is the header length, and n is the number of packets to be transmitted. Because of the employment of ARQ, the LLC entity at the receiving side will continue to send requests to its peer for retransmission of the discarded packet, until it is correctly received or the retransmission counter has reached a certain number. Hence, the total number of transmitted bits N_{Total} comprises the number of bits to be transmitted, plus the number of bits retransmitted. Assume the packet error rate is P_P and the maximum number of retransmissions is set to N_R , then

$$\begin{aligned} N_{Total} &= nL_P + nL_P P_P + nL_P P_P^2 + \dots + nL_P P_P^{N_R} \\ &= \frac{nL_P(1 - P_P^{N_R})}{1 - P_P} \end{aligned} \quad (3.16)$$

Therefore, Eq. (3.15) can be rewritten as

$$S_{MAC} = \frac{1 - P_P}{1 - P_P^{N_R}} \cdot R \quad (3.17)$$

where,

$$R = \frac{L_P - H_P}{L_P} \quad (3.18)$$

If the packet error probability P_P is small and N_R is large, then the term $P_P^{N_R}$ can be ignored, and thus the throughput can be approximated as

$$S_{MAC} \cong (1 - P_P)R. \quad (3.19)$$

As shown in Eq. (3.8), P_P is also a function of P_B , which, as given by Eq. (3.12), is also a function of the bit error probability P_e . Therefore, the throughput can be estimated by the channel bit error rate. As an example, suppose that the packets are transmitted in their default length, i.e. 1,088 bits per packet, the relationship between P_P and P_B in the AWGN channel is shown in Eq. (3.6), then the corresponding expression for the throughput is,

$$S_{MAC} = \frac{L_P - H_P}{L_P} \cdot \left[(1 - P_B)^4 \cdot \frac{9}{274} + (1 - P_B)^5 \cdot \frac{265}{274} \right] \quad (3.20)$$

where L_P is 1,088 bits, H_P is 48 bits, and P_B is expressed in Eq. (3.12).

3.5 Numerical Results and Discussion

3.5.1 BLERs and PERs

Fig. 3.4 shows that for an AWGN channel with coherent detection, our computer simulation results of BLER and PER as a function of E_b/N_o match the curves derived from Eq. (3.12) and (3.8) respectively. According to [1], the parameters in Eq. (3.12) are $N = 63$, $K = 47$, $t = 8$, and $l = 6$. The same figure also shows that the difference between the BLER and the PER is about 0.25~0.5 dB. For example, to achieve 1% block error rate the required E_b/N_o is 4.25 dB, while it requires 4.6 dB to achieve 1% packet error rate.

However, the performance difference in a fading environment tends to be much more significant. Using the computer simulation model described in Chapter 2, below we present our simulation results generated in a slow Rayleigh fading channel with several Doppler shift values for a 1-bit differential receiver. The simulation for BLER/PER generation has tested from a minimum of 2,072 to a maximum of 576,000 blocks. Each block is 368 bits long and each bit is

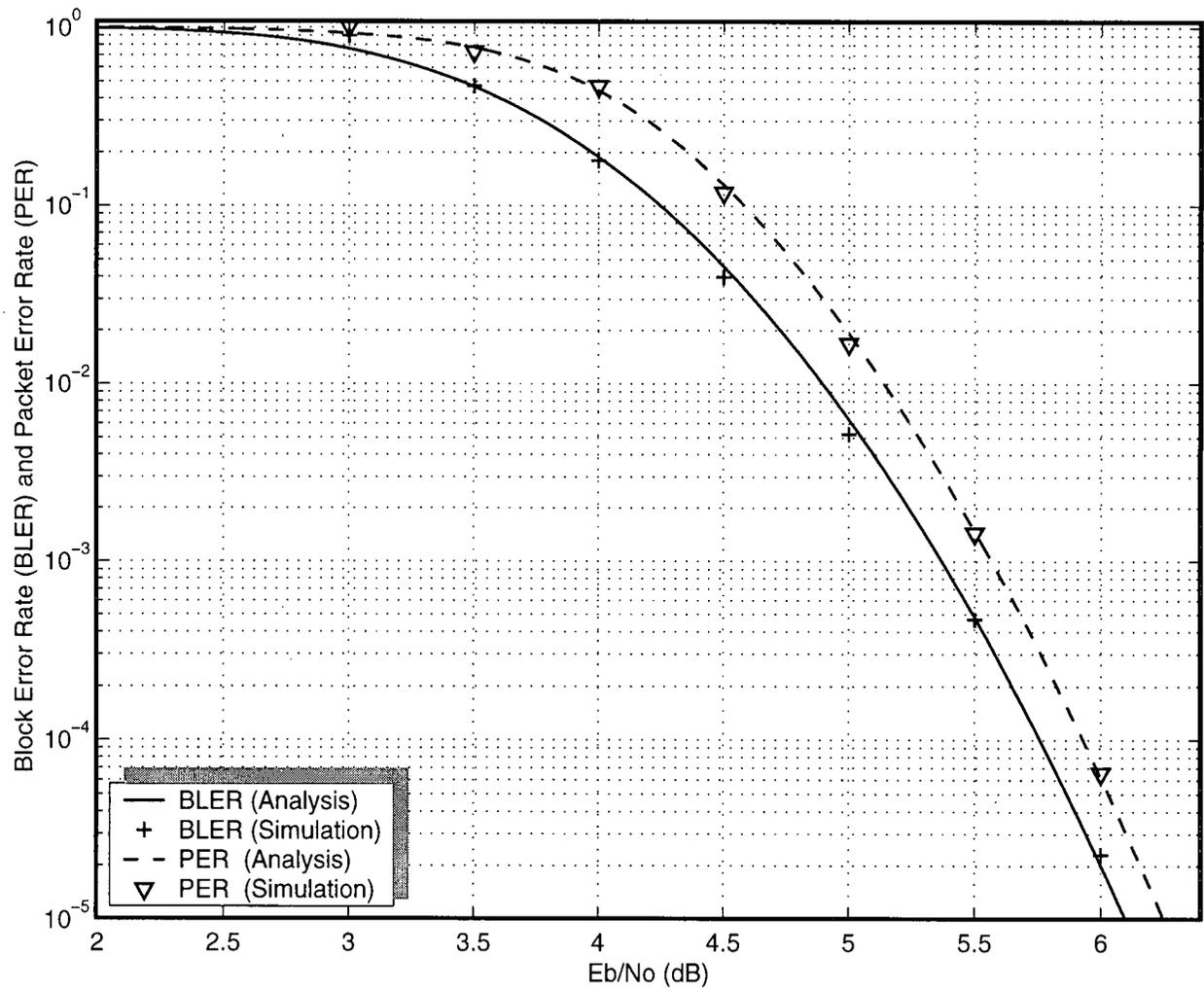


Figure 3.4 BLER and PER in the AWGN channel

sampled at eight samples/bit. The simulation results were obtained by means of Monte Carlo error counting techniques. In order to have accurate performance results for all simulated BLER/PER values, at least 100 erroneous blocks were counted for BLER from 10^{-1} to 10^{-3} and at least 50 erroneous blocks for BLER of 10^{-4} . The confidence level of at least 90% is obtained for all simulation results, with the confidence intervals listed in Table 3.1 for each BLER. The intervals are calculated by using the formula in [30] (Eq. (9)).

BLER	Number of Blocks Tested	90% Confidence Interval
10^{-1}	2,072	(9.22E-2, 1.08E-1)
10^{-2}	8,240	(8.80E-3, 1.14E-2)
10^{-3}	111,950	(8.96E-4, 1.12E-3)
10^{-4}	576,000	(8.58E-5, 1.17E-4)

Table 3.1 The 90% confidence interval for each BLER

Figs. 3.5-3.7 illustrate the BLER and PER performance versus E_b/N_o with 1-bit differential detection for Doppler shift $f_D=20$ Hz, 40 Hz and 80 Hz, respectively. For comparison purposes, we also consider coherent detection with ideal carrier recovery³. Compared to the results for ideal coherent detection, the results for differential detection are only approximately 5 dB worse, which implies that differential detection is relatively robust in the fading environment. As it shown in terms of E_b/N_o , the BLER declines faster than the PER, which is worse than the BLER about 2.5 to 5 dB with various Doppler shifts. It is easy to observe that with Rayleigh fading channel the PER is almost unacceptable for a wide range of E_b/N_o .

³ An ideal carrier recovery means that the carrier signal is correctly extracted at the receiver. *However, it must be emphasized that this is an ideal case which is practically impossible. It is used as an upper bound for performance comparison purposes.*

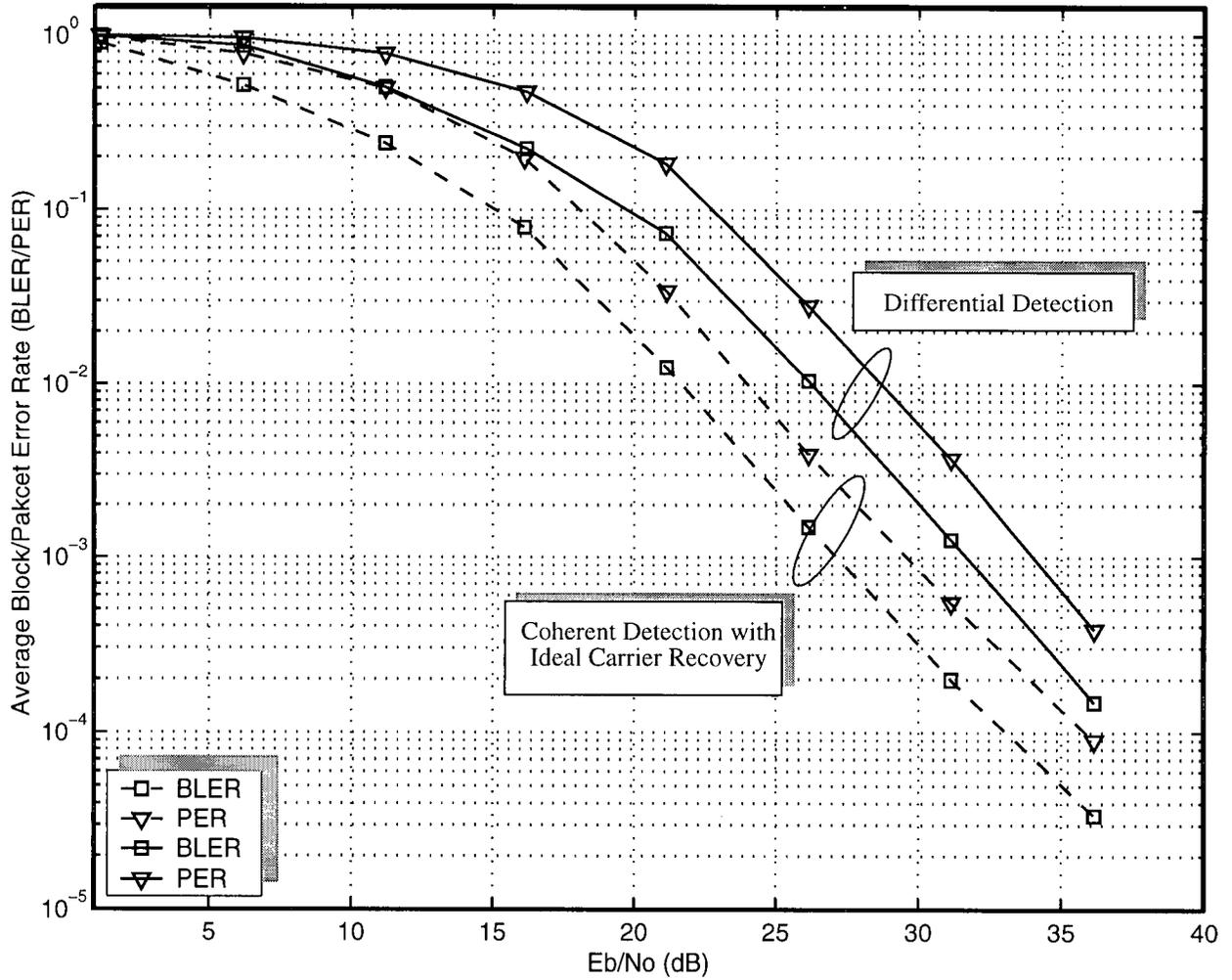


Figure 3.5 BLER and PER with differential detection in the Rayleigh fading channel with $f_D = 20$ Hz

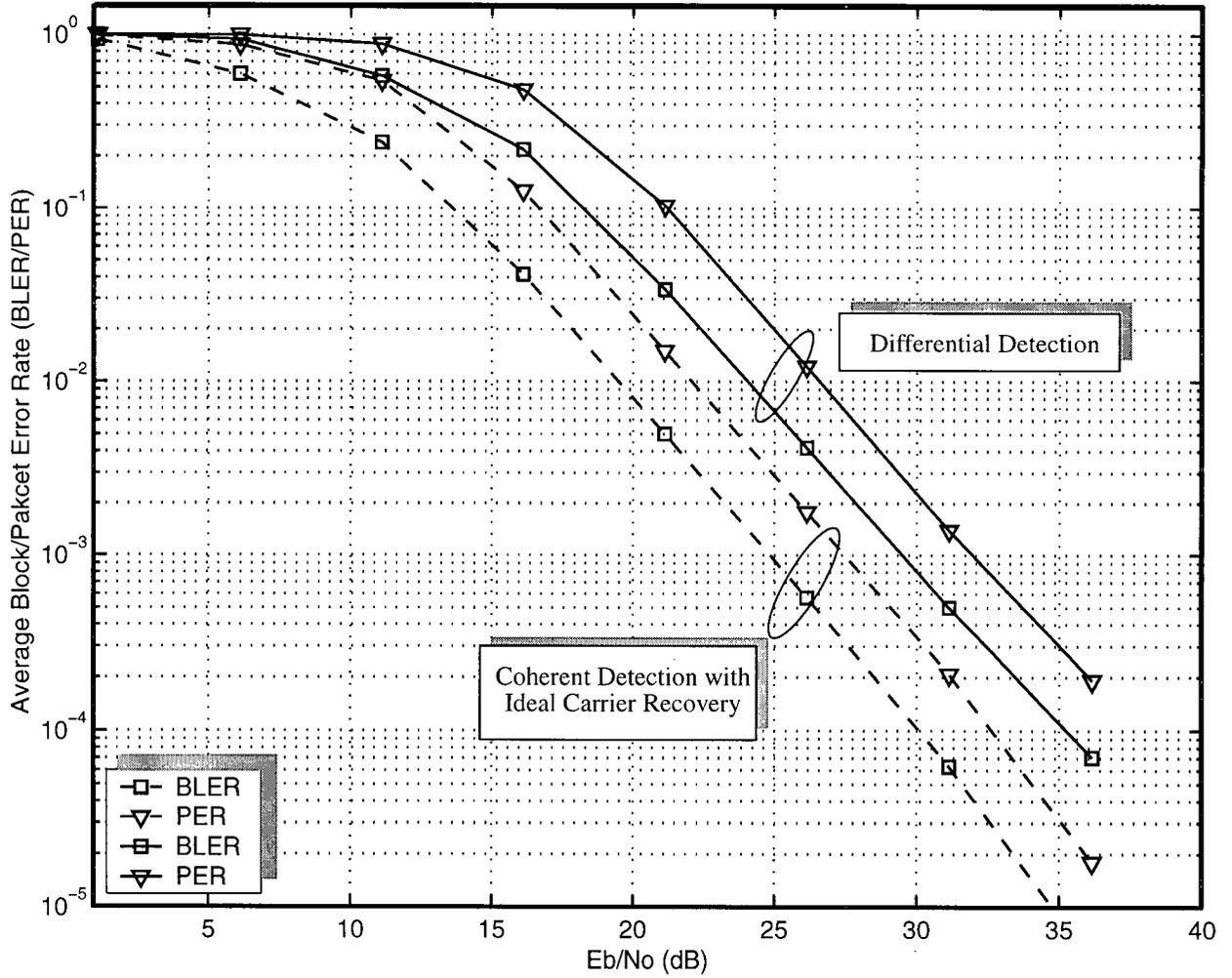


Figure 3.6 BLER and PER with differential detection in the Rayleigh fading channel with $f_D=40$ Hz

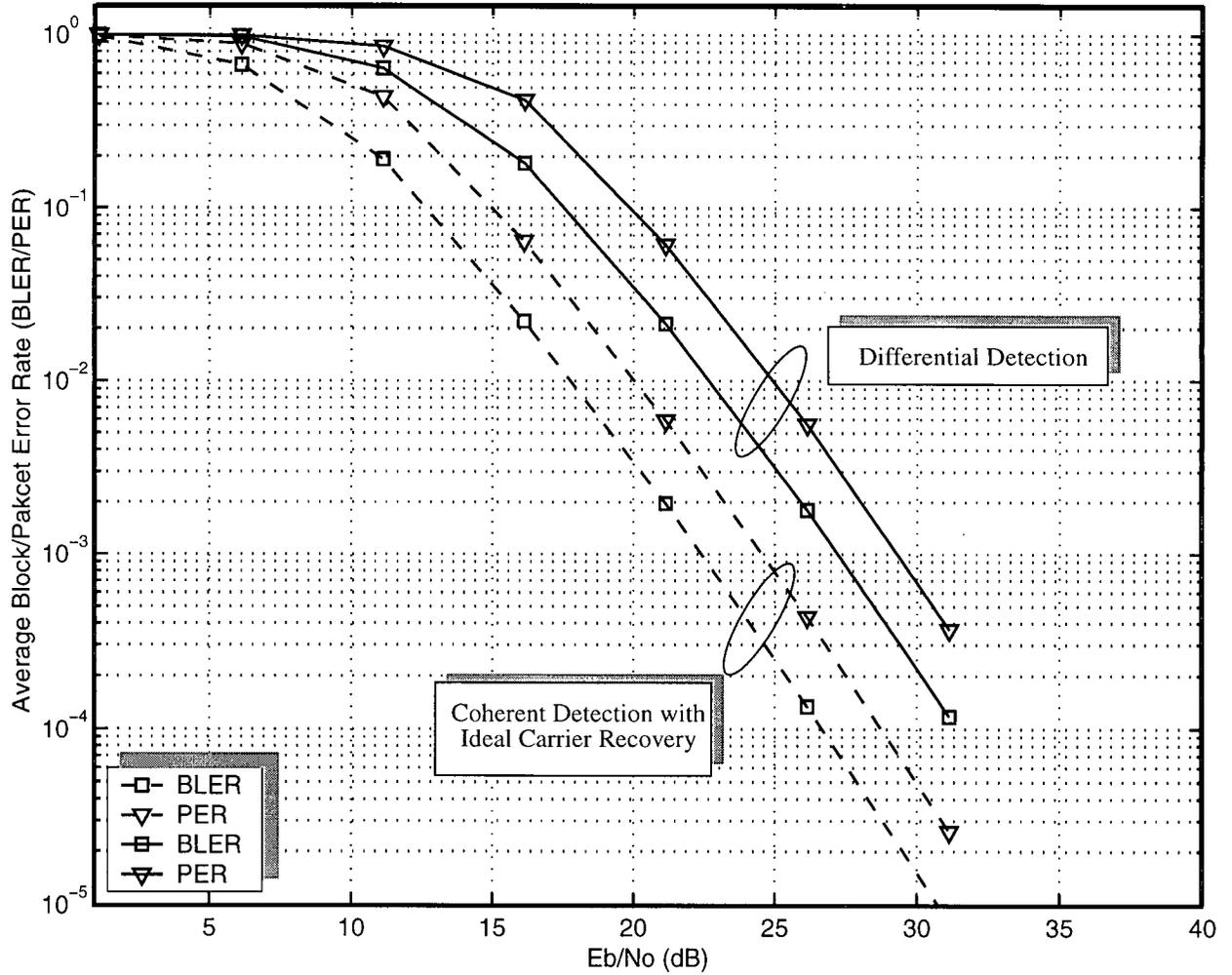
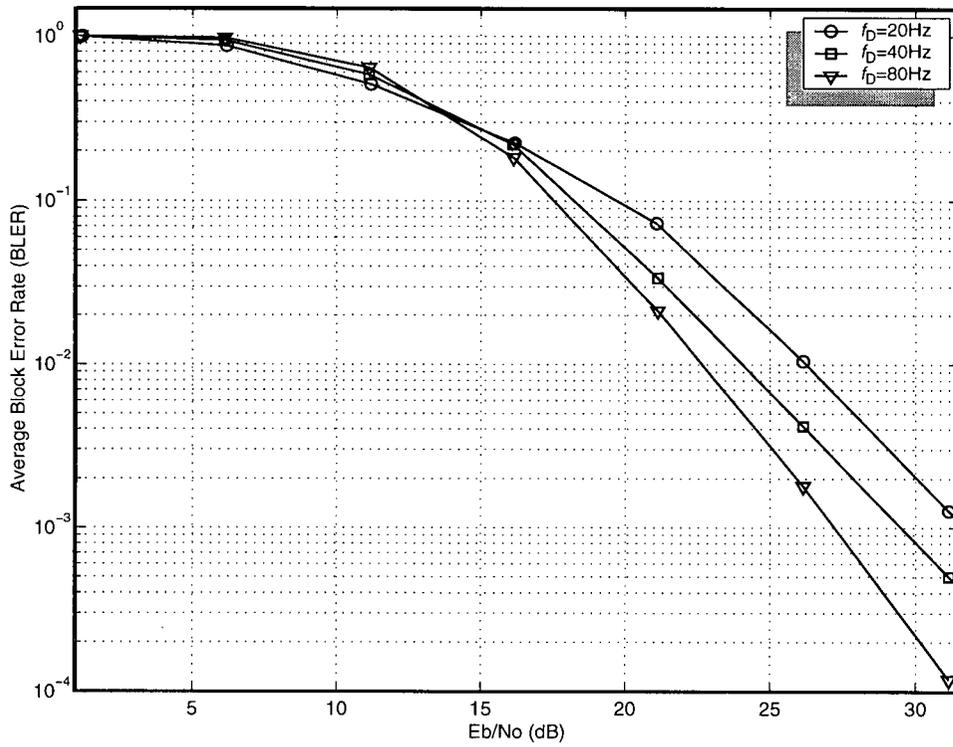
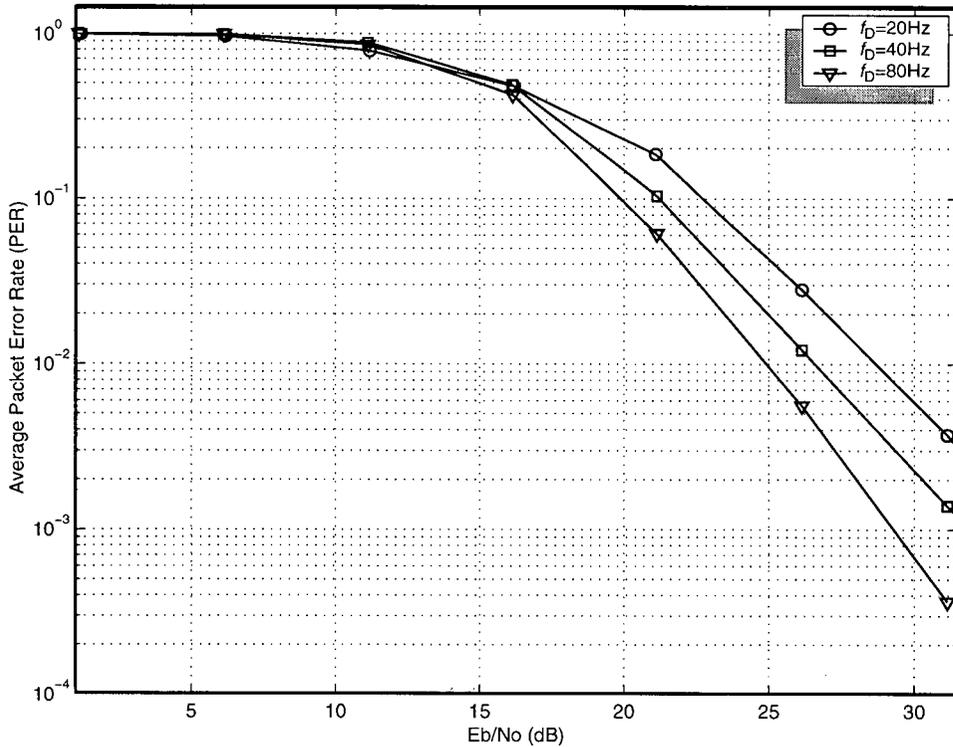


Figure 3.7 BLER and PER with differential detection in the Rayleigh fading channel with $f_D = 80$ Hz

By comparing the performance evaluation results presented in Figs. 3.5-3.7, we also found it interesting that the impact of different Doppler shift values on the block/packet error performance is different. Fig. 3.8 gives a more clear view of the BLER/PER for different f_D s. As it shown, the BLER/PER decreases as the Doppler shift f_D increases from 20 Hz to 80 Hz. For example, when the E_b/N_o is approximately 26 dB, the BLER is around 10^{-2} when $f_D = 20$ Hz, while it is around 10^{-3} when $f_D = 80$ Hz. Another point of view of this phenomenon is to look at the bit error rates with the employment of error correction codes for different Doppler shifts, i.e. the estimated probability of bits in error after the error correction. As it shown in Fig. 3.9, the BER for $f_D = 80$ Hz is as much as 5 dB better than the BER for $f_D = 20$ Hz as the E_b/N_o is large. This applies for both differential detection and the ideal coherent detection. The phenomenon can be justified by the argument that the error correction scheme employed in CDPD performs better in a less slower fading environment. This is because of the characteristics of the fading channel and the error correction scheme. In a slow fading environment, the long duration of deep fades causes some blocks to contain a large number of errors while others contain none (or a lot fewer) errors. The packet containing these large number of errors will have higher probability to be uncorrectable. However, this is not the case for less slower fading, where all blocks encounter errors but the number is relatively smaller. Hence, the error correction scheme is more efficient when f_D is large due to the larger spread of errors. For example, with RS-coding scheme employed in CDPD system, an RS-block will be uncorrectable if it contains less than 8 erroneous codewords. It means the number of erroneous bits has to be large enough and spread enough to cover 8 codewords. Assume $E_b/N_o = 20$ dB, if $f_D = 20$ Hz, there are approximately 91.5% blocks containing 8 or less erroneous codewords, and hence they are correctable. If $f_D=80$ Hz, however,



(a)



(b)

Figure 3.8 Comparison of BLERs and PERs with different Doppler shift frequencies with differential detection in the Rayleigh fading channel. (a) BLER. (b) PER.

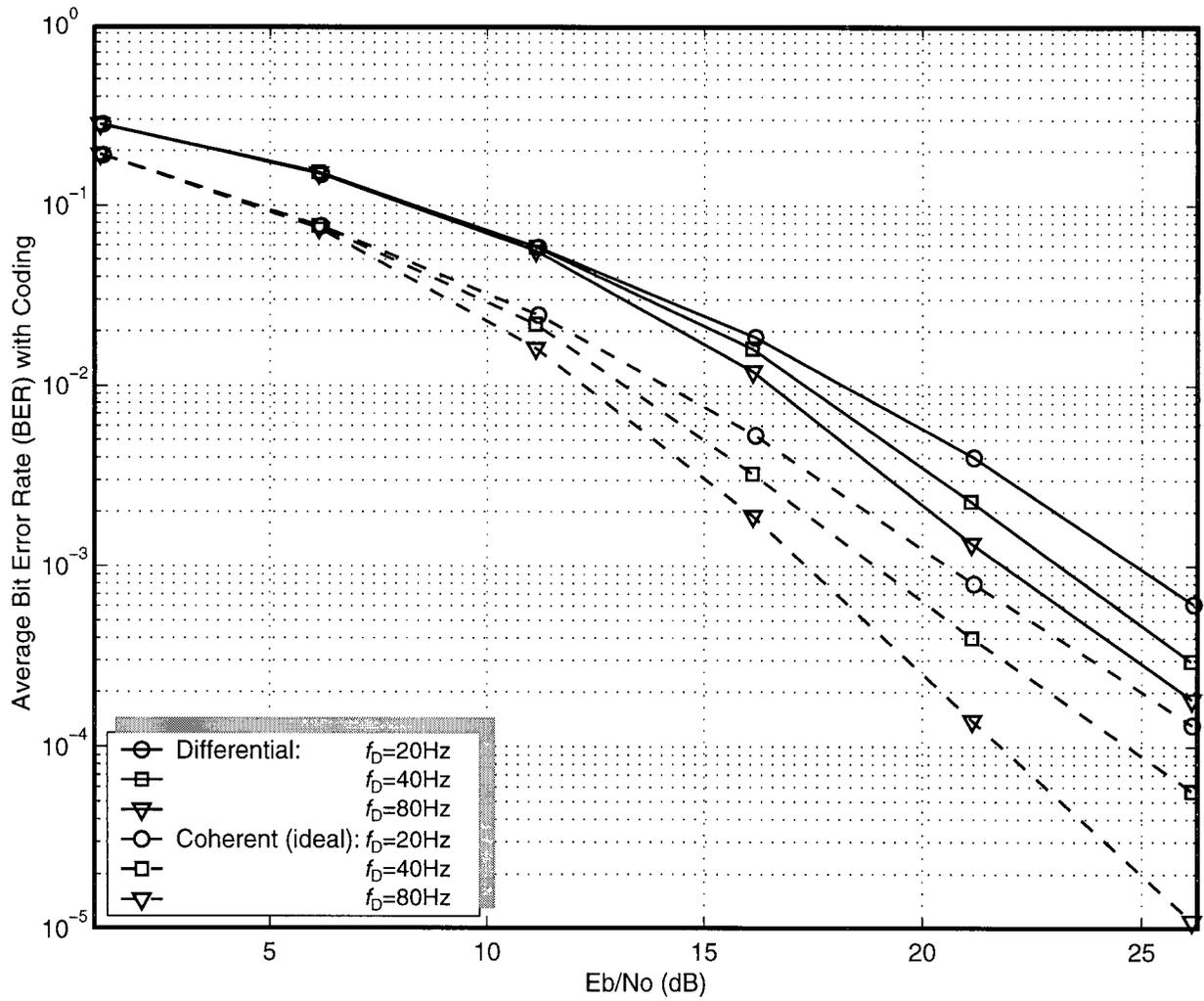


Figure 3.9 BER with error correction coding in the Rayleigh fading channel.

there are approximately 96.5% blocks which are correctable. On the other hand, when the fading becomes faster, i.e. when f_D increases to a certain large value, or a threshold, the number of errors contained in the blocks will exceed the error correction capability, and the block error performance will start to decline. Our simulation result indicates that when f_D reaches 400-450Hz, the BLER/PER starts to increase and the degradation becomes very significant when f_D is larger than about 500 Hz.

3.5.2 CUD

The percentage of CUD with differential detection in the fading environment is presented in Fig. 3.10, where three Doppler shifts $f_D = 20$ Hz, 40 Hz, and 80 Hz are considered. Analogous to the results in AWGN channel, the three percentage curves tend to be zero in very low and very high ranges of E_b/N_o s, but the peaks of the curves are approximately between 36%~43%. The percentages for $f_D = 20$ Hz has the highest value of 43%, while the peak for $f_D = 80$ Hz is almost 36%. Compared to the case for AWGN channel, the curves for fading channel cover a much wider range of E_b/N_o s. This implies that over a relatively wide range of E_b/N_o s the system is working in an inefficient manner. For example, the percentage of CUD is over 20% for nearly 10-dB range of E_b/N_o s. These curves are generated with the packet length configurations described in Chapter 2, i.e. test data traffic contains SP, MP and LP with equal probability. Although packet length is a parameter affecting the amount of CUD, according to our experiments, the CUD with different settings of packet sizes varying from the minimum to the maximum LPDU lengths, exhibits similar percentage peaks to those shown in Fig. 3.10. Evidently, it is an source of inefficiency that requires improvements.

Comparing the peak values of the CUD with the three Doppler shifts, we also found it

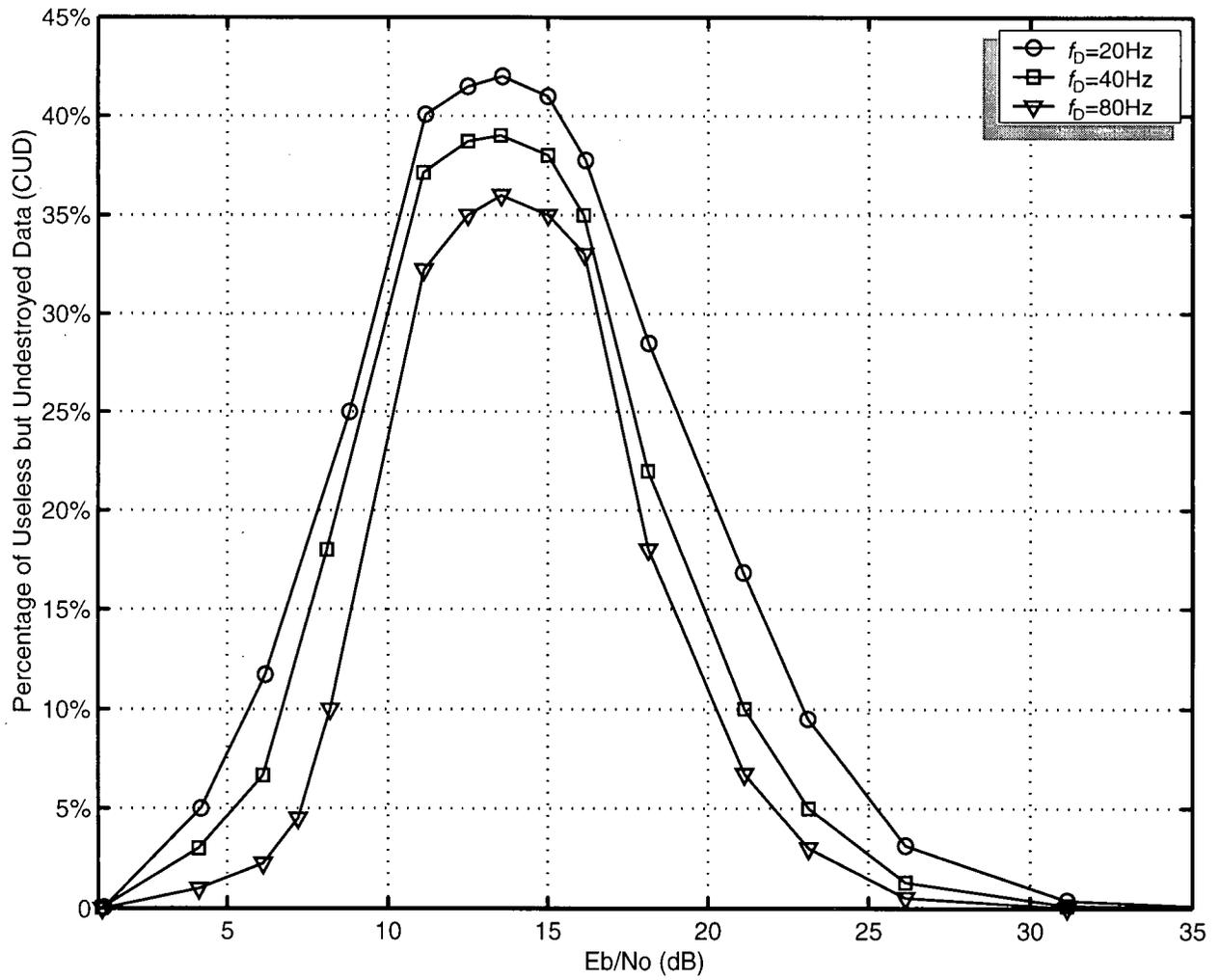


Figure 3.10 Percentage of CUD with differential detection in the Rayleigh fading channel.

interesting that, although the difference between the three f_D s are significant, i.e. 80 Hz is four times of 20 Hz, the values of the CUD do not show remarkable difference. For example, the peak CUD percentage with $f_D = 20$ Hz is about 43%, while that with $f_D = 80$ Hz differs only 7% as 36%. According to the observation we can argue that the differential detection in a Rayleigh fading channel is relatively insensitive to the change of the Doppler shifts.

3.5.3 Throughput

In Section 3.4, we have discussed the relation between the throughput and the BLER, and further the relation between the throughput and the BER. Eq. (3.20) has shown an expression of the throughput as a function of P_B , given that the packets are transmitted in their default lengths, i.e. 1,088 bits per packet. Substituting Eq. (3.12) into Eq. (3.20), the throughput can be represented as a function of the BER, the plot of which is illustrated in Fig. 3.11. As confirmed by the simulation results, the throughput increases as the BER decreases, and it approaches zero as the BER increases to above $3.0\text{E-}2$, whereas it approaches its maximum value when the BER drops below $7.0\text{E-}3$. Due to the packet overhead, the throughput can never be 1, even if there is no packet loss at all. Instead, the maximum throughput, corresponding to the ideal transmission, i.e. $P_P = 0$, can be computed as $(L_P - H_P)/L_P \cong 0.96$.

Let us consider the throughput in terms of the channel conditions, i.e. as a function of the E_b/N_o . Simulation results have shown that similar curves can be derived for the Rayleigh fading channel with random packet lengths. Fig. 3.12 illustrates the throughputs versus the E_b/N_o in Rayleigh channel with Doppler shifts $f_D = 20$ Hz, 40 Hz, and 80 Hz. The solid lines corresponds to the results for differential detection, and the dashed lines show that the throughput for ideal coherent detection is as much as 30% more than the differential detection. Note that the results for

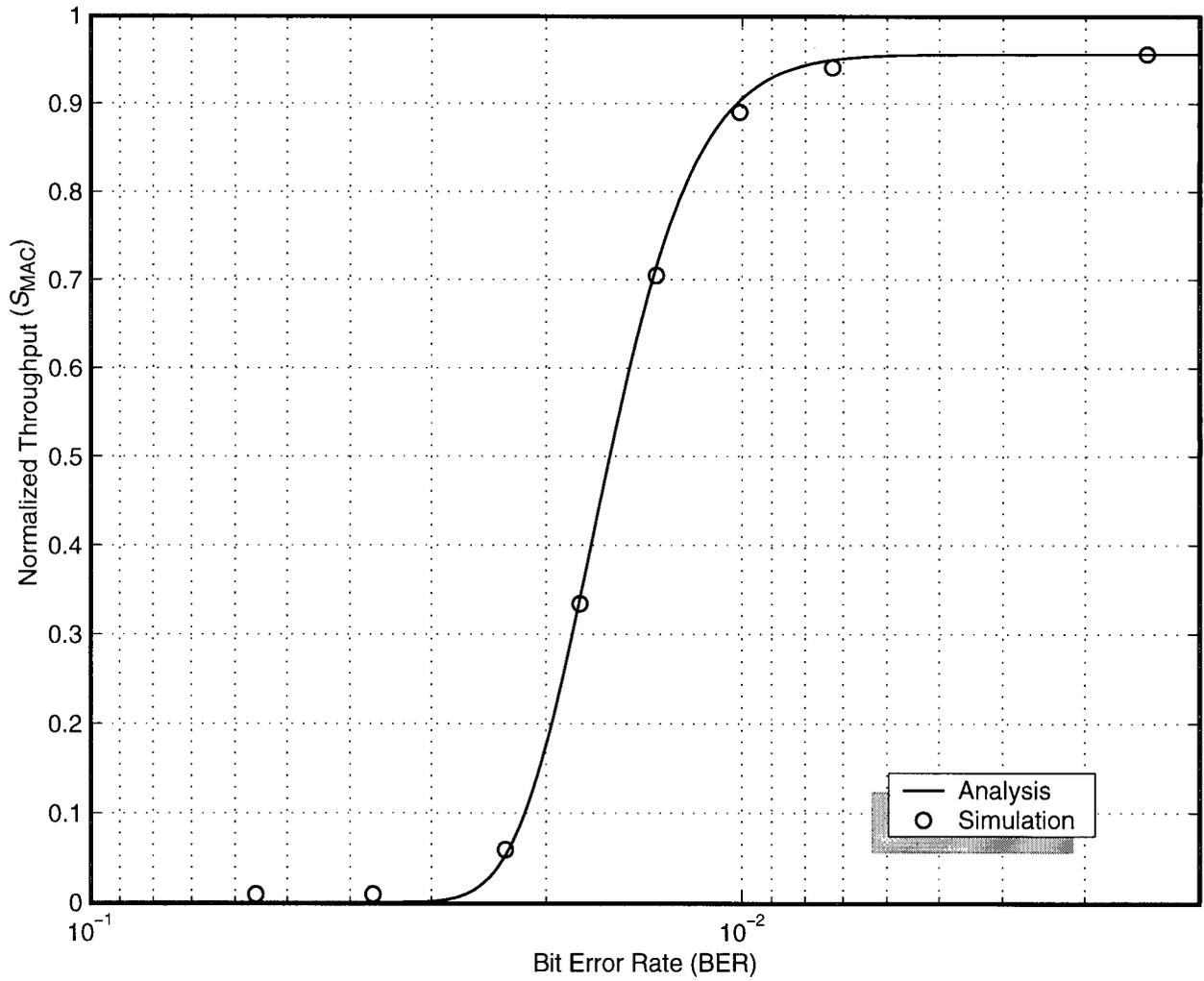


Figure 3.11 Throughput versus BER with coherent detection in the AWGN channel, where the packets are transmitted in their default lengths.

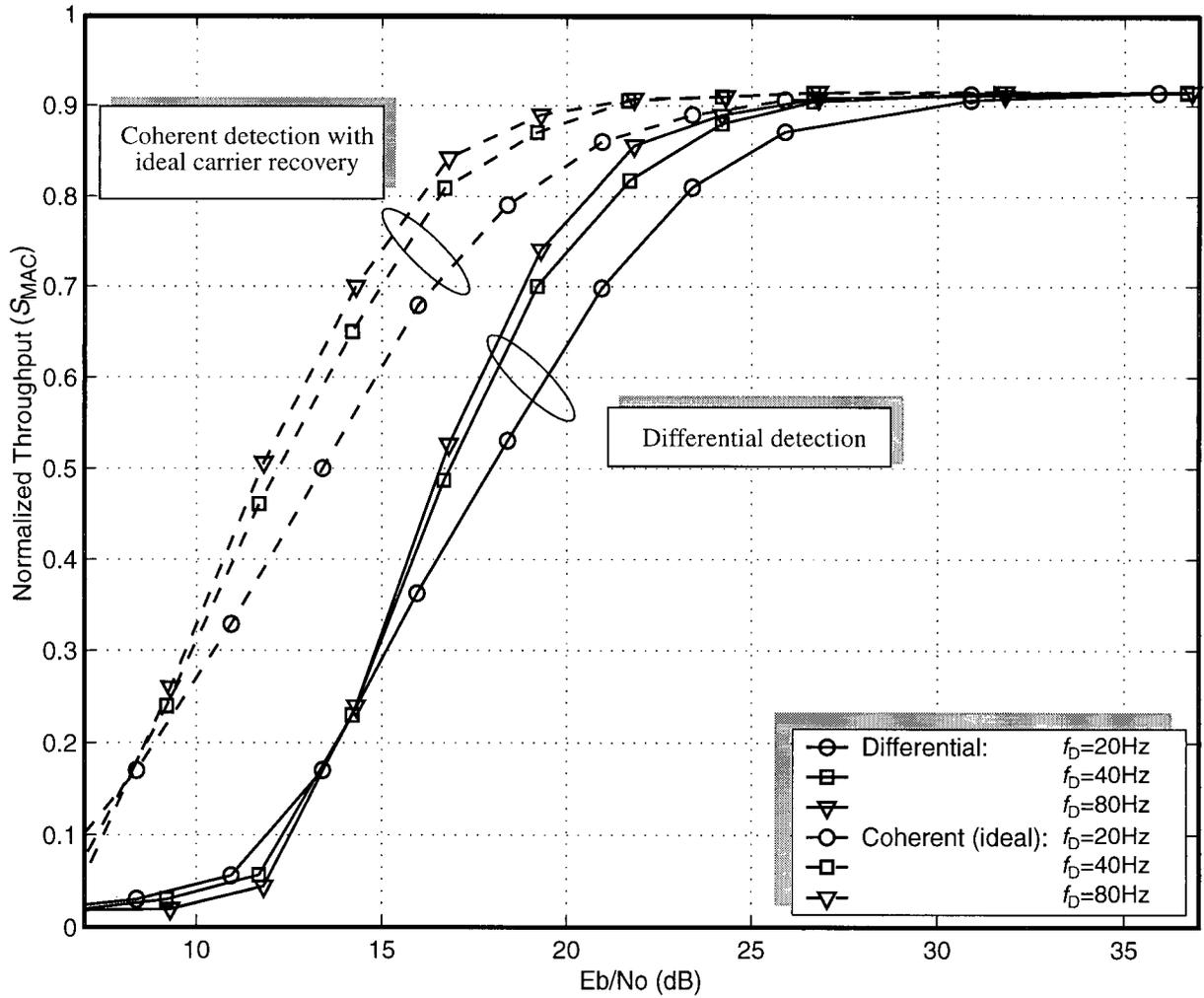


Figure 3.12 Throughput versus E_b/N_0 in the Rayleigh fading channel.

ideal coherent detection is an unrealistic case and should be only considered as the throughput upper bound for comparison purposes. For all the three Doppler shifts, the throughputs increase as the E_b/N_o increases. The throughputs for smaller Doppler shifts are slightly less than those for larger Doppler shifts when the E_b/N_o is between 14 and 30 dB. However, as the E_b/N_o increases to a very large value, e.g. larger than 35 dB, the curves for three Doppler shifts tend to merge at approximately 0.92, which is a little less than the throughput with fixed packet length (0.96) because the transmission of fixed default-length packets yields more throughput than the transmission of unfixed random-length packets.

3.6 Summary

In this chapter we have investigated analytically and by means of computer simulations, the system performance in terms of the RS-block error rate, the packet error rate, and the throughput at the MAC layer. The relationship between the BLER and the PER is analyzed and the CUD is identified as a source of performance inefficiency. The percentage of CUD has been investigated by means of computer simulation. This chapter has also presented the analytical results of the BLER, PER, and the throughput in the AWGN channel, and the corresponding simulation results for both coherent and differential detector in AWGN channel and a slow Rayleigh fading channel. It has been realized through the performance study that the system can be improved if the source of degradation can be eliminated.

Chapter 4 MAC-ARQ Protocol

4.1 Introduction

As discussed in the previous chapter, it is the correct but useless data (CUD) that causes the degradation of the CDPD throughput performance. In order to mitigate the problem, in this chapter we will propose a new scheme that effectively overcomes the aforementioned deficiencies. The proposed scheme employs an ARQ protocol that operates at the MAC layer, i.e. a protocol that requests retransmission of the erroneous data blocks instead of LPDUs at the LLC layer. It will be referred to as the MAC-ARQ protocol. In this way, only the unrecoverable information is retransmitted and no CUD data arises. A similar approach is applied in the General Packet Radio Service (GPRS) [31][32], where the Radio Link Control (RLC) protocol that operates below the data link layer handles the block transmission and correction. The organization of this chapter is as follows. After this introduction, the description and performance evaluation for the new protocol will be presented in Section 4.2, where both its advantages and performance tradeoff will be discussed. In the next section, Section 4.3, we will propose an adaptive scheme which maximizes the throughput over the conventional and the MAC-ARQ scheme. Finally, a summary will be contained in Section 4.4.

4.2 Description and Performance Evaluation of the MAC-ARQ Protocol

In Chapter 3, it was mentioned that for the CDPD system, conventional ARQ is operating in the LLC layer. It only provides retransmission of LPDUs in this layer. Unlike conventional ARQ, MAC-ARQ is an auto-retransmission scheme operating in the MAC layer, by providing retransmission of RS-blocks in MAC layer, hence avoids retransmission of the LPDUs. As a consequence, the effect of the CUD could be eliminated. For the retransmission, each block needs

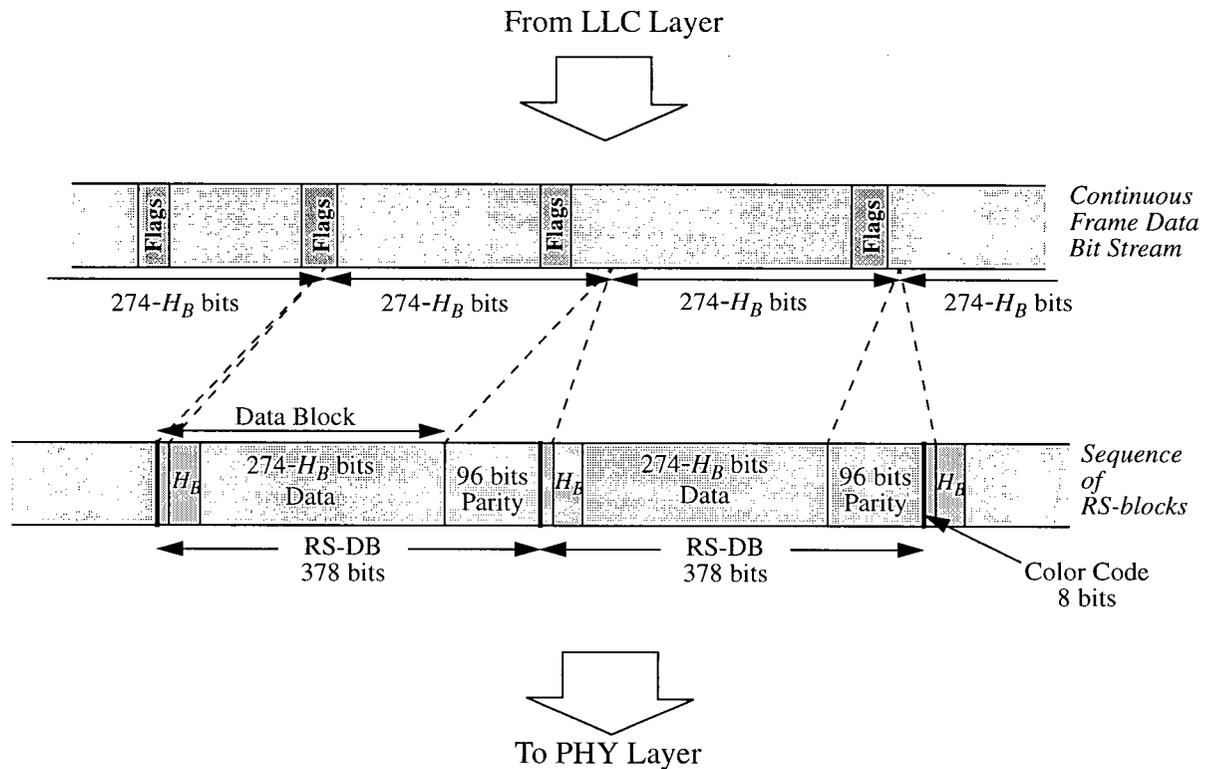


Figure 4.1 Data transmission procedures in the forward channel.

to contain necessary information for its destination and its sequence number. The description of the MAC-ARQ is as follows. As illustrated in Fig. 4.1, the sequence of packets formed by the LPDUs is segmented into $(274 - H_B)$ bits each, where H_B is the length of the header to be attached to each data block. Then each segment is prefixed by a block header (H_B), together with the 8-bit Color Code, yielding a 282-bit data block. The data block is thereafter RS encoded and transmitted to the physical layer. The block header contains the destination address(s) and the block sequence number to perform the ARQ operations. The destination address(s) identifies the logic link connection(s) where this MAC-block contains data from. Since a block may contain data from more than one logic link connection, the address field needs to contain addresses for all the connections. Therefore, an implementation dependent algorithm which performs address

compression and mapping can be applied to ensure the MES side to be able to identify the data block destined to itself from this address field. The receiving MES identifies the sequence numbers of the blocks which fail decoding, and sends request(s) via the reverse channel for the retransmission of these blocks, while queues the correctly received blocks which have not complete an LPDU. Upon receipt of the retransmitted blocks, the MES assembles an LPDU with these blocks as well as the queued blocks, and forward it up to the LLC layer. Therefore, at the LLC layer, there will be no missing packets and thereby no retransmission involved.

Since all the erroneous data blocks are corrected at the MAC layer, the probability of a packet in error will be zero. Referring to the simulation results for BLERs and PERs in the fading channel, e.g. Figs. 3.5-3.7, the up to 5-dB of performance degradation between the PER and the BLER is eliminated by the MAC-ARQ protocol.

The penalty we pay for the performance improvement this new protocol offers, is that extra redundancy needs to be embedded into the forward channel, i.e. the header of each block, and therefore the performance is expected to decline under very good channel conditions. Following Eq. (3.15), the MAC layer throughput for the MAC-ARQ protocol is

$$S'_{MAC} = \frac{N_I}{N'_{Total}} = \frac{N_{Total}}{N'_{Total}} \cdot S_{MAC} \quad (4.1)$$

where N'_{Total} is the actual number of bits transmitted for MAC-ARQ. Similarly to the derivation of Eq. (3.19), the ratio N_{Total}/N'_{Total} can be expressed as

$$\frac{N_{Total}}{N'_{Total}} \cong (1 - P_B)R', \quad (4.2)$$

where

$$R' = \frac{L_B - H_B}{L_B} \quad (4.3)$$

where L_B is 274 bits, and H_B , containing the necessary destination address and block sequence number, can be sufficiently assumed as three octets or 24 bits in our system. From Eqs. (3.19) and (4.2), Eq (4.1) can now be expressed as

$$S'_{MAC} \cong RR'(1 - P_P)(1 - P_B) . \quad (4.4)$$

Since the MAC-ARQ protocol has ensured every LPDU can be successfully decoded, i.e. $P_P=0$, Eq. (4.4) can be rewritten as

$$S'_{MAC} \cong RR'(1 - P_B) . \quad (4.5)$$

Substituting Eq. (3.12) into Eq. (4.5) yields the throughput expression in terms of the bit error probability, and this is plotted in Fig. 4.2. The dashed line in Fig. 4.2 illustrates the throughput versus the P_e of the new scheme in an AWGN channel, where the packets are transmitted only in their default lengths. As compared to the conventional scheme, the MAC-ARQ scheme has up to 45% higher throughput when P_e is larger than the cross point $1.15E-2$, while the conventional scheme outperforms the new one by nearly 10% when P_e is very small, i.e. after P_e crosses the threshold.

The simulation performance evaluation results for the throughput versus E_b/N_o in Rayleigh fading channels are presented in Fig. 4.3. The results reflect the throughput while packets are transmitted in random lengths. For comparison purposes, the results for the conven-

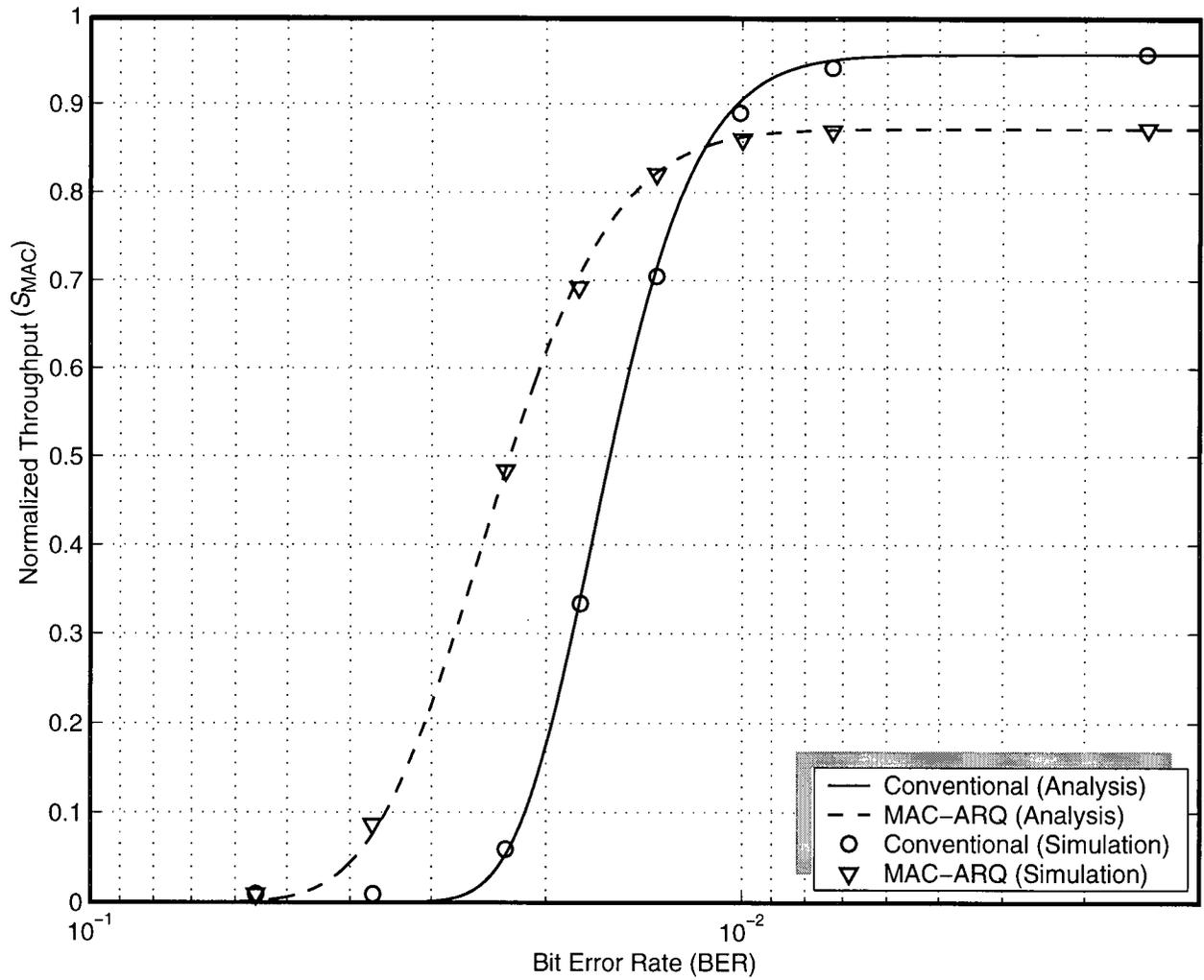


Figure 4.2 Throughput versus BER with coherent detection in the AWGN channel, where the packets are transmitted in their default lengths.

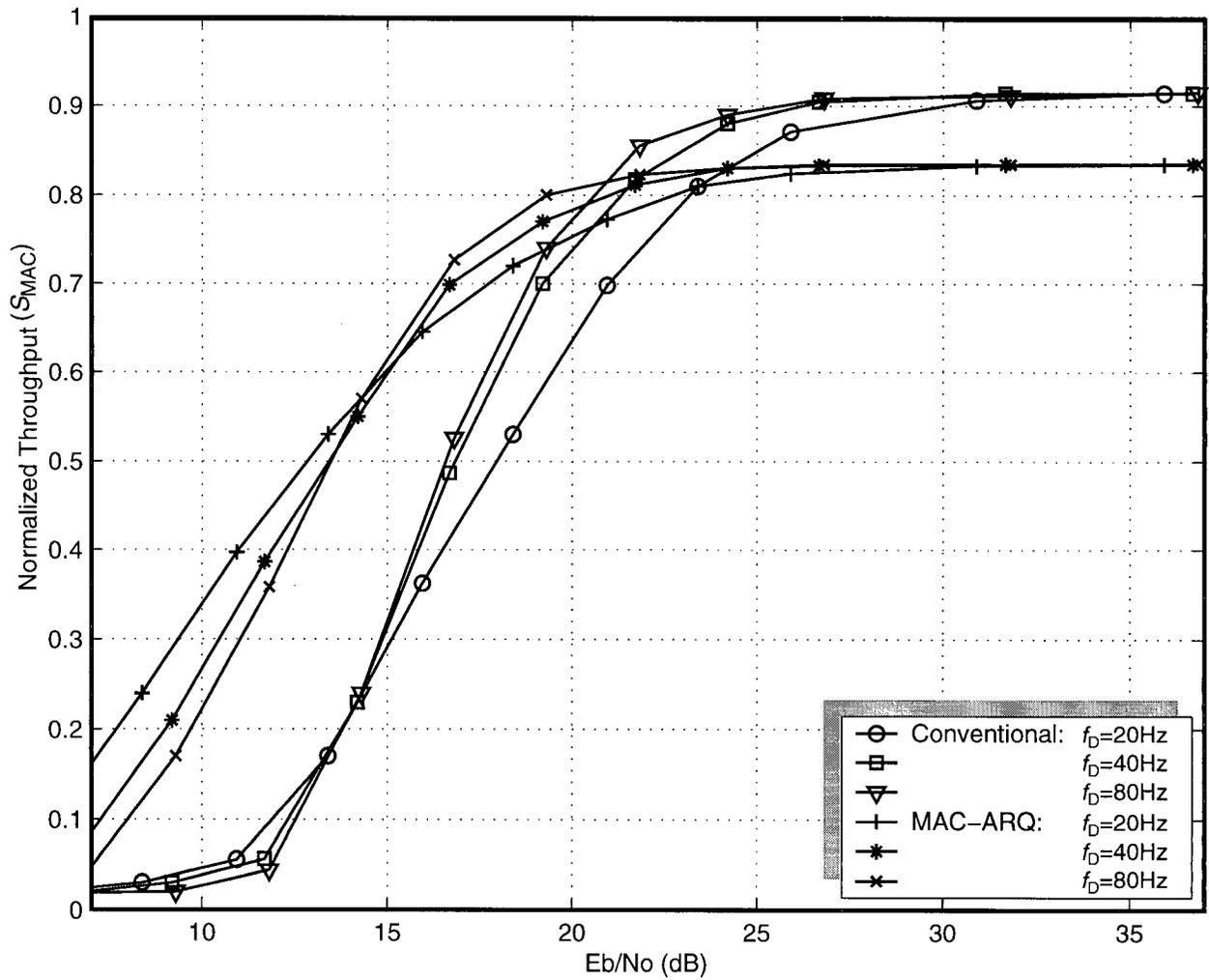


Figure 4.3 Throughput versus E_b/N_0 with differential detection in Rayleigh fading channel, where the packets are transmitted in their random lengths.

tional scheme are also included. As it shown in the figure, the MAC-ARQ scheme shows over 30% increase of the throughput as E_b/N_o is low (e.g. $E_b/N_o < 15$ dB), while exhibits approximately 10% degradation as E_b/N_o is large (e.g. $E_b/N_o > 30$ dB). The cross points of these two schemes are ranged within 21~23 dB. As discussed in the previous chapter, the MAC throughput can be estimated by the channel BER performance. Figs. 4.4-4.6 illustrate the simulation results for the throughput versus bit error rate in Rayleigh fading channels. Similar P_e thresholds can be identified in these figures. Let the curve for $f_D = 80$ Hz be an example, when P_e is larger than the threshold, where $P_{e,threshold} = 7.4E-3$, the MAC-ARQ scheme has up to 35% higher throughput than the conventional scheme. Nevertheless, the new scheme has about 10% lower throughput than the conventional one when P_e is smaller than the threshold. The throughputs for other Doppler shifts vary but they exhibit similar thresholds. The observed thresholds are listed in Table 4.1 below.

f_D (Hz)	BER Threshold
20	5.0E-3
40	6.5E-3
80	7.4E-3

Table 4.1 Observed BER thresholds for various Doppler shifts

4.3 The Adaptive Scheme

In this section we will discuss an Adaptive scheme which can make further performance improvements, by means of dynamically enabling the ARQ operation at either MAC layer or the LLC layer. Similar to some previous work concerning adaptive ARQ scheme, such as [37][38], where the blocks are transmitted in various sizes as the changing of the channel, the adaptive ARQ protocol that we are proposing, will simply switch the operations according to the channel

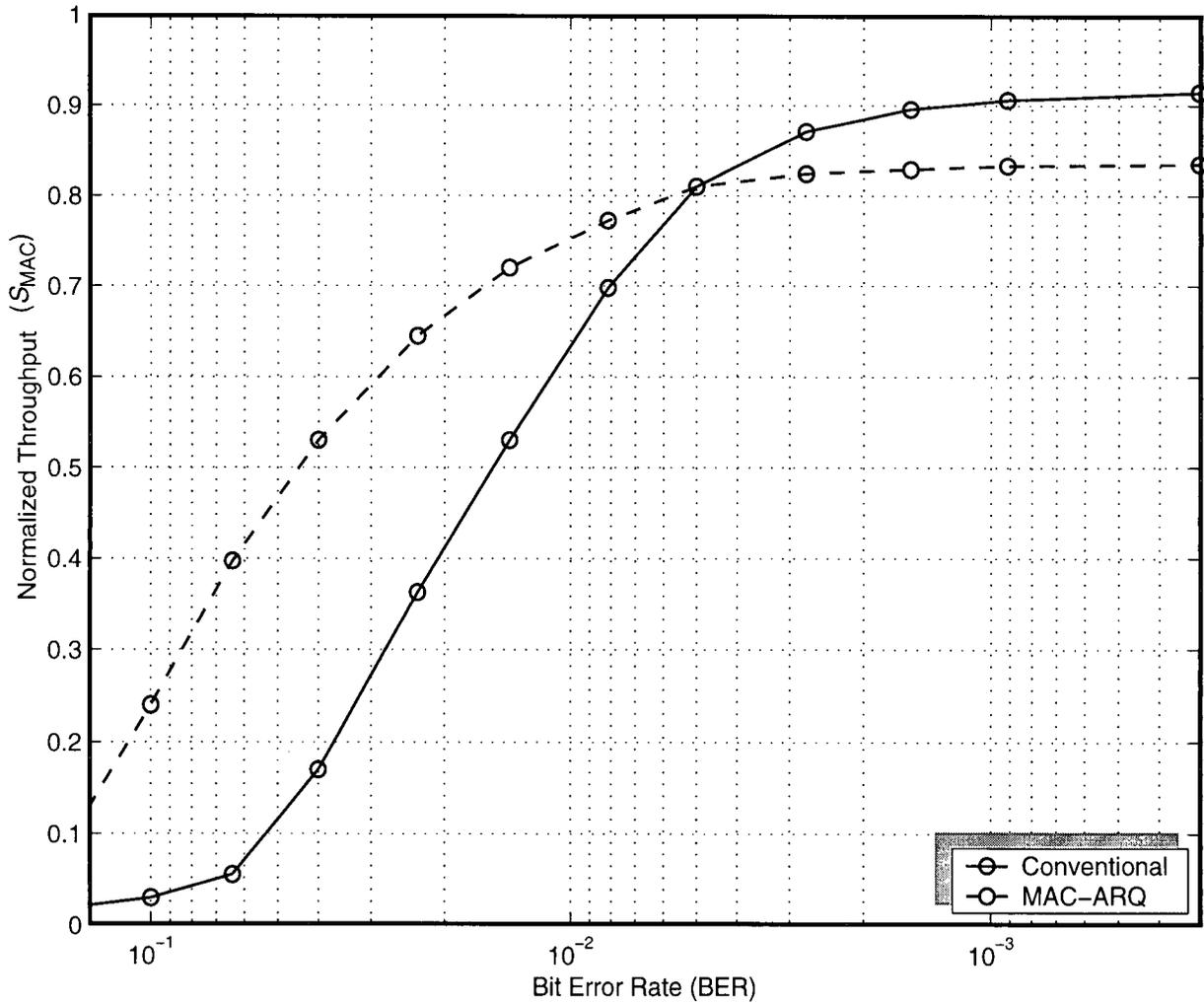


Figure 4.4 Throughput versus BER with differential detection in Rayleigh fading channel ($f_D=20\text{Hz}$), where the packets are transmitted in their random lengths.

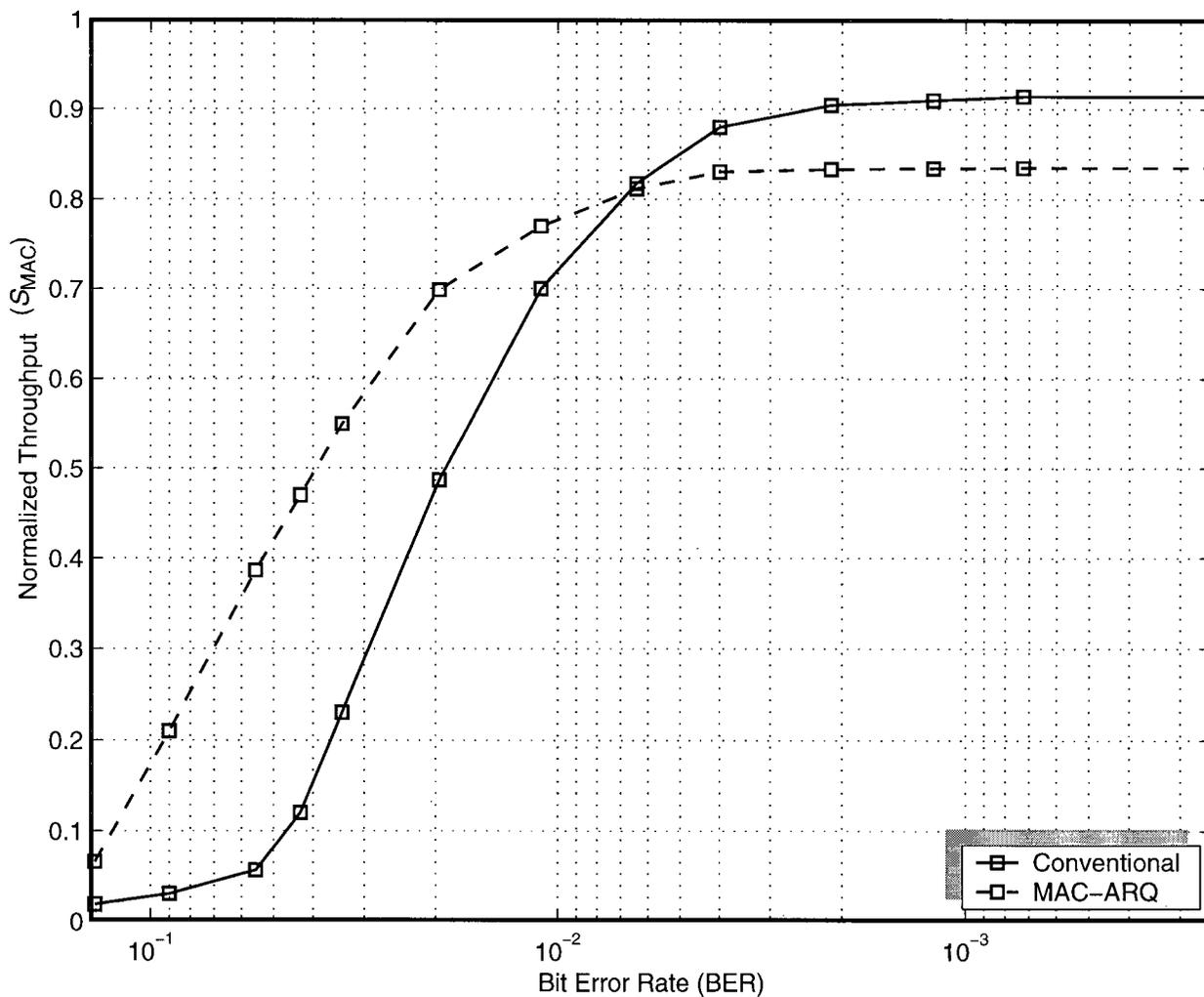


Figure 4.5 Throughput versus BER with differential detection in Rayleigh fading channel ($f_D=40\text{Hz}$), where the packets are transmitted in their random lengths.

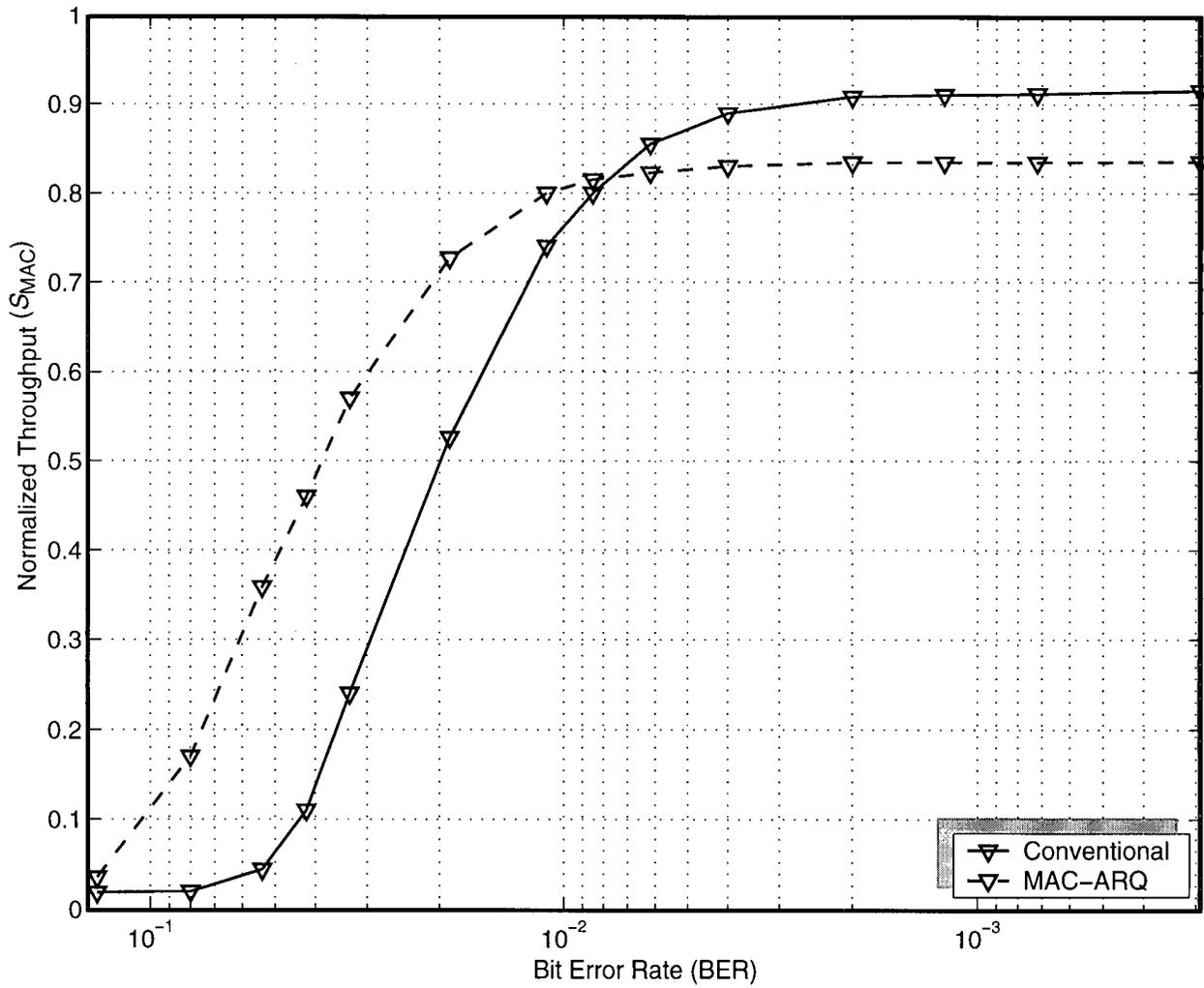


Figure 4.6 Throughput versus BER with differential detection in Rayleigh fading channel ($f_D=80\text{Hz}$), where the packets are transmitted in their random lengths.

conditions. Since it is not a complicated task to estimate the channel BER between the MDS and the M-ESs, we can take advantage of the relations between the throughput and the BER to optimize the throughput performance. For example, as shown in the previous section, in various fading conditions, the two schemes exhibit similar BER thresholds whereas the throughput of one scheme outperforms another on one side of the threshold and vice versa on another side. Hence, it is worth considering to employ an adaptive scheme where the MAC-ARQ protocol can be automatically enabled and disabled according to the estimated BER performance. In other words, enable the MAC-ARQ scheme when the BER is larger than the threshold, and disable it when BER is smaller than the threshold. For example, the base station will start to transmit data with MAC-ARQ protocol, where each data block to be transmitted from the base station to the M-ES is prefixed by a block header before it is RS encoded. During the transmission, the M-ES constantly estimates the channel condition and BER performance. When the BER reaches up to the threshold, the M-ES may send an LLC control frame via the reverse channel to request disabling the MAC-ARQ protocol. Upon the reception of the control frame, the base station will switch to the conventional mode, where the data blocks are transmitted without the block header. If the BER drops below the threshold again, the base station may enable the MAC-ARQ protocol again upon the request by the M-ES. Therefore, if we ignore the processing time of the adaptive operation, this scheme will maintain the throughput performance in its maximum value range. Let us take $f_D = 80\text{Hz}$ as an example, when the channel BER is less than the threshold, e.g. $\text{BER} = 3.0\text{E-}2$, the system will take the MAC-ARQ scheme and provide a throughput of 0.6, which is almost twice the throughput provided by the conventional scheme. On the other hand, when BER is larger than threshold, e.g. $\text{BER} = 1.0\text{E-}3$, the system will take the conventional scheme instead and provide a throughput of 0.91, which is around 0.1 more than what it can provide by the MAC-ARQ scheme.

4.4 Summary

In this chapter we have proposed, analyzed, and evaluated by means of computer simulations, the performance of the MAC-ARQ scheme. We have presented the advantages on throughput performance of the MAC-ARQ protocol as well its tradeoff in certain channel conditions. We have also identified the performance threshold which can be represented by channel BER performance. Consequently, we have discussed an adaptive scheme which automatically select between the two schemes according to the estimated BERs, thus to achieve optimal throughput.

Chapter 5 Conclusion and Some Suggestions for Future Research

5.1 Conclusions

In this thesis we have studied, analyzed, and evaluated CDPD forward channel performance, in terms of MAC layer block error rate, packet error rate, and the throughput. Our studies have included both analysis and computer simulations, in the presence of various communication channel conditions, including the AWGN channel and land-mobile Rayleigh fading channel. We have also taken into account different type of GMSK receivers, such as the coherent receiver and differential receiver. Computer simulation results have shown that coherent receiver demonstrates better performance than the differential receiver in AWGN channel while it is outperformed by the differential receiver in the fading channel.

We have investigated the forward channel MAC layer structure and identified a source of performance degradation, i.e. the correct but unusable data (CUD) which peaks about 45% over all transmitted data regardless the channel conditions. To eliminate this inefficiency source, we have proposed an MAC-ARQ protocol which performs ARQ operations at the block level, hence effectively eliminates the CUD. Simulation results have illustrated considerable performance improvement of this protocol. In the fading channel, it demonstrates over 0.3 of increase of the throughput when E_b/N_o is relatively small, although when E_b/N_o is very high, the extra overhead introduced by the MAC-ARQ protocol results in approximately 0.1 degradation on the throughput. When studying the relations between throughput and channel BERs over various channel criteria, we have found that the throughput curves for the two schemes intersect at similar BER values. In order to optimize the throughput performance, we have discussed an adaptive scheme

which dynamically switches back and forth between the conventional scheme and the proposed scheme in accordance with the physical channel BER performance. By this means the maximum throughput can be achieved.

5.2 Suggestions for Future Research

5.2.1 Effect of the MAC-ARQ protocol to the reverse channel

This thesis has not studied the effect of the MAC-ARQ protocol to the reverse channel performance. Since the ARQ is operating at the MAC layer, it is possible that the number of acknowledgment packets transmitted on the reverse channel is more than that in a conventional scheme; this may cause degradation on the reverse channel performance. Therefore, it would be worth looking into such effects on the reverse channel.

5.2.2 Revoke the ARQ operation at the data link layer for the MAC-ARQ scheme

The proposed scheme performs ARQ operation at the MAC layer, while the original ARQ functionality still exists in the upper layer. The MAC-ARQ protocol requires an extra block header containing necessary information for the ARQ, but in the LLC layer the conventional ARQ also requires similar header, this causes some redundancy. For the sake of simplicity, this thesis has not taken this redundancy into consideration. Therefore, it might be useful to perform a small modification on the LLC layer protocol to revoke the ARQ operation, and hence eliminates the redundancy.

5.2.3 Soft threshold for the adaptive scheme

Although this thesis has demonstrated similar BER thresholds in the throughput studies over various channel criteria, the BER threshold cannot be estimated accurately since the variety of channel conditions, e.g. for different Doppler frequencies, the BER thresholds are slightly

different. It would be interesting to consider a soft threshold which would involve some statistical studies of channel characteristics to determine the threshold range where the throughput is maximized.

5.2.4 Performance study for similar applications other than CDPD

There are many wireless communication applications that possess similar system structure as CDPD. They may have different block/packet lengths and overhead. It would be interesting to investigate the performance for other configurations, and verify the possibility that the MAC-ARQ protocol and the adaptive scheme being applied to these systems and evaluate the improvements of the performance.

5.2.5 Diversity reception

In this thesis we have investigated the CDPD performance with coherent and differential detection with single antenna reception. For better performance in a fading environment, diversity reception should be a good choice. The combination of diversity reception and the MAC-ARQ/Adaptive scheme will produce significant performance improvement.

5.2.6 More advanced coding techniques

CDPD system employs (63,47) Reed-Solomon coding scheme as the FEC. It provides fairly strong error correcting capability. However, in this thesis we have found that the RS codes performs better in a slower fading environment than in a faster fading environment. It would be interesting to investigate the effects of the codes in fading channel with different Doppler shift values, and introduce more advanced coding techniques which give better performance in a slower fading environment.

References

- [1] CDPD Forum, *Cellular Digital Packet Data System Specification, Release 1.1*, January 1995.
- [2] M. Sreetharan and R. Kumar, *Cellular Digital Packet Data*, Artech House, Inc., 1996.
- [3] A. K. Salkintzis, "Packet Data Over Cellular Networks: The CDPD Approach," *IEEE Commun. Mag.*, vol. 37, no. 6, Jun. 1999, pp. 152-159.
- [4] A. K. Salkintzis and P. T. Mathiopoulos, "Adaptive Beamforming in CDPD Mobile End Systems," *6th IEEE International Conf. on Electronics, Circuits and Systems*, Pafos, Cyprus, 5-8 Sept. 1999.
- [5] D. Saha and S. E. Kay, "Cellular Digital Packet Data Network," *IEEE Trans. Veh. Technol.*, vol. 46, no. 3, Aug. 1997, pp. 697-706.
- [6] W. A. Massey and R. Srinivasan, "A packet delay analysis for Cellular Digital Packet Data," *IEEE J. Select. Commun.*, vol. 15, no. 7, Sep. 1997, pp. 1364-1370.
- [7] B. Kamali, "Modulation and channel coding for Cellular Digital Packet Data networks," *Applied Microwave & Wireless Mag.*, vol. 9, no. 3, May/June 1997, pp. 26-38.
- [8] K. C. Budka, "Cellular digital packet data: Channel availability," *IEEE Trans. Veh. Technol.*, vol. 46, no. 1, Feb. 1997, pp. 31-40.
- [9] C. Jedrzycki and V.C.M. Leung, "Optimal channel assignments strategies for forced hopping in CDPD systems," *J. Mobile Networks and Applications*, vol.5, no.1, 2000, pp. 5-16.
- [10] T. Wan and A. U. Sheikh, "Performance evaluation of Cellular Digital Packet Data (CDPD) systems," *48th IEEE Veh. Techn. Conf.*, Ottawa, Ont., 18-21 May 1998, vol. 2, pp. 810-814.
- [11] S. Resheff, "CDPD Performance: A Joint Analysis of the Physical and MAC Layers," *8th IEEE International Symposium, PIMRC '97*, vol. 2, Sept. 1997, pp.440-446.

- [12] C. Leung and J. NG, "Estimation of block error rates for NCFSK modulation on a VHF/UHF mobile radio channel," *IEEE Trans. Veh. Technol.*, vol. 38, no. 2, May 1989, pp. 46-49.
- [13] H. Bischl and E. Lutz, "Packet error rate in the non-interleaved Rayleigh channel," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, Feb./Mar./Apr. 1995, pp. 1375-1382.
- [14] A. Seyoum and N. C. Beaulieu, "Semianalytical simulation for evaluation of block-error rates on fading channels," *IEEE Trans. Commun.*, vol. 46, no. 7, Jul. 1998, pp. 916-920.
- [15] A. S. Tenenbaum, *Computer Networks*, 3rd ed., Prentice Hall, 1996.
- [16] K. Murota and K. Hirade, "GMSK modulation for digital mobile radio telephony," *IEEE Trans. Commun.*, vol. 29, July 1981, pp. 1044-1050.
- [17] J. S. Toor, "Differentially detected GMSK systems in the presence of adjacent channel interference and nonlinearities," *Master's Thesis*, Dept. of Electrical and Computer Engineering, UBC, Jul. 1994.
- [18] M. K. Simon and C. C. Wang, "Differential detection of Gaussian MSK in a mobile radio environment," *IEEE Trans. Veh. Technol.*, vol. 33, Nov. 1984, pp. 307-320.
- [19] S. S. Shin, "Differentially detected MSK and GMSK modulation schemes in CCI channels for mobile cellular telecommunications systems," *Master's Thesis*, Dept. of Electrical and Computer Engineering, UBC, Oct. 1992.
- [20] C.-E. W. Sundberg, "Continuous phase modulation," *IEEE Commun. Mag.*, vol. 24, Apr. 1986, pp. 25-38.
- [21] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, New York, 1965.
- [22] W. C. Jakes Jr., *Microwave Mobile Communications*, Wiley, New York, 1974.
- [23] W. C. Y. Lee, *Mobile Cellular Telecommunication Systems*, McGraw-Hill, New York, 1989.
- [24] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, 1989.

- [25] S. S. Shin and P. T. Mathiopoulos, "Differentially detected GMSK signals in CCI channels for mobile cellular telecommunication systems," *IEEE Trans. Veh. Techn.*, vol. 42, no. 3, Aug. 1993, pp. 289-293.
- [26] A. Yongacoglu, D. Makrakis, and K. Feher, "Differential detection of GMSK using decision feedback," *IEEE Trans. Commun.*, vol. 36, no. 6, Jun. 1993, pp. 641-649.
- [27] C. W. Lee, Y. M. Chung, and S. U. Lee, "Bit error probabilities of CPM signals in frequency-selective fading channels," *IEE Proceedings-I*, vol. 138, no. 5, Oct. 1991, pp. 465-472.
- [28] K. Hirade, F. Adachi, K. Ohtani, and M. Ishizuka, "Error-rate performance of digital FM with differential detection in land mobile radio channels," *IEEE Trans. Veh. Techn.*, vol. 28, no. 3, Aug. 1979, pp. 204-212.
- [29] W. Stallings, *Data and Computer Communications*, 4th ed., Prentice Hall, 1994.
- [30] M. C. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE J. Sel. Areas Commun.*, vol. 2, Jan. 1984, pp. 153-170.
- [31] G. Brasche and B. Walke, "Concepts, Services, and Protocols of the New GSM Phase 2+ General Packet Radio Service," *IEEE Commun. Mag.*, vol. 35, no. 8, Aug. 1997, pp. 94-104.
- [32] ETSI, "General packet radio service (GPRS); Service description," *GSM 03.60*, Version 7.1.0, 1998.
- [33] S. B. Wicker, *Error Control Systems for Digital Communications and Storage*, Prentice Hall, 1995.
- [34] B. Kamali and R. L. Brinkley, "Reed-Solomon coding and GMSK modulation for Digital Cellular Packet Data systems," *IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing, PACRIM*, Victoria, BC, 20-22 Aug. 1997, vol. 2, pp. 745-748.
- [35] W. C. Y. Lee, *Mobile Communications Design Fundamentals*, Wiley, New York, 1993.
- [36] P. Lettieri and M. B. Srivastava, "Adaptive frame length control for improving wireless link throughput, range, and energy efficiency," *Proceedings IEEE INFOCOM'98 Conf. on Computer Communications Seventeenth Annual Joint Conf. of the IEEE Computer and*

- Communications Societies Gateway to the 12st Century*, San Francisco, USA, 29 Mar.-2 Apr. 1998, vol. 2, pp. 564-571.
- [37] S. Hara, A. Ogino, M. Araki, M. Okada, and N. Morinaga, "Throughput performance of SAW-ARQ protocol with adaptive packet length in mobile packet data transmission," *IEEE Trans. Veh. Techn.*, vol. 45, no. 3, Aug. 1996, pp. 561-569.
- [38] J. A. C. Martins and J. D. C. Alves, "ARQ protocols with adaptive block size perform better over a wide range of bit error rates," *IEEE Trans. Commun.*, vol. 38, no. 6, June 1990, pp. 737-739.
- [39] W. Ajib and P. Godlewski, "Acknowledgment operations in the RLC layer of GPRS", *MoMuC '99, 1999 IEEE International Workshop on*, 15-17 Nov. 1999, pp. 311-317.
- [40] D. Kostas, "Proposed MAC ARQ Mechanism for the proposed 802.16.4 Standard", *IEEE 802.16 Broadband Wireless Access Working Group* <<http://www.ieee802.org/16/arc/802-16-tg4/msg00026.html>>, IEEE 802.16.4-01/21, Mar. 2001.

Appendix A. Program Listings

```

global.h
*****
/* Name: Global.h
/* Description: Global include file and variable
/* definitions
*****
#ifndef GLOBAL_H
#define GLOBAL_H
/*
/* Included files
/*-----
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
/*
/* Global definitions
/*-----
#define T 1 /* Symbol period
#define S 8 /* Samples per symbol
#define PI 3.14159265
#define LLEN 1088 /* Large LPDU length
#define MLEN 568 /* Medium LPDU length
#define SLEN 48 /* Small LPDU length
#define INFLEN 274 /* Info bits in an RS-block
#define PARLEN 96 /* Parity bits in RS-block
#define BLEN 378 /* Length of an RS-block
#define ERRMAX 8 /* RS-code correctability
#define DPCOUNT_MAX 100 /* Max number of LPDUs
/* for random number generators
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define NTAB 32
#define NDIV (1+(IM-1)/NTAB)
#define EPS 1.2e-7
#define RMX (1.0-EPS)
/*-----
/* Global variable definitions
/*-----
extern longsymbols, /* number of symbols
L; /* number of samples, = symbols*s
extern longiduma[2]; /* seeds for rand numbers, I&Q channel
extern long*idum[2];
extern double ENR;
extern const char ch_type = 'A'; /* A: AWGN, F: Rayleigh fading
extern const char dt_type = 'C'; /* C: Coherent, D: Differential
extern const double fd = 80.0; /* Hz, doppler freq for fading
extern double Eb_ch; /* Energy per bit
extern long t_fade_init; /* initial of fade vector
typedef struct Outputs {
double ENR;

```

```

double Eb_ch;
long t_count; /* total number of err bits after correction
long t_badInfBits_PHY; /* total number of bad bits in PHY
long t_badInfBits_DL; /* total number of bad bits in Datalink
long t_badPacs; /* total number of bad packets
long t_badBlks; /* total number of bad blocks
long t_infBits; /* total number of inf bits transmitted
long t_biks; /* total number of blocks transmitted
long t_pacs; /* total number of packets transmitted
} outs;
extern FILE *fp;
#endif

cdpmain.c
*****
/*
/* cdpmain.c
/*
/*-----
#include "global.h"
/* Name: main()
/* Description: main function
/* Input: None
/* Output: None
/*-----
void main()
{
const int points = 1;
const double ENRV [points] = {7.5};
int count;
int doneFlag;
int doLoada[2], doSavea[2];
int *doLoad[2], *doSave[2];
long int tmp1 = -1*(1+rand()), tmp2 = -1*(1+rand());
for (int i=0; i<points; i++)
{
memset (&outs, 0, sizeof(outs));
outs.ENR = ENRV[i];

iduma[0] = tmp1; iduma[1] = tmp2;
doLoada[0] = 0; /* 0: don't load initial setting, 1: load
doLoada[1] = 0;
doSavea[0] = 0; /* always initial to 0
doSavea[1] = 0;
idum[0] = &iduma[0]; idum[1] = &iduma[1]; /* for channel.c
doLoad[0] = &doLoada[0]; doLoad[1] = &doLoada[1];
doSave[0] = &doSavea[0]; doSave[1] = &doSavea[1];
t_fade_init = 1;
Eb_ch = 0.0;
doneFlag = 0;
symbols = 0;
L = 0;

```

```

symbols = ((long)(ceil(((double)(dpv_len)/INFLEN))) * BLEN);
L = symbols * s;
outs.t_infBits += dpv_len;
outs.t_pacs += DPCOUNT_MAX+1;
outs.t_blks += symbols/BLEN;
)

```

dlAsm.c

```

/*****
*/
/* dlAsm.c
*/
/*****
#include "global.h"
long symbols, /* number of symbols */
L; /* number of samples, = symbols*s */
void dl_assemble_frames();
/*****
*/ Name: dl_assemble_frames()
/* Description: assemble LPDUs and calculate missing
*/ frames
/* Input: None
/* Output: None
/*****
void dl_assemble_frames()
{
char frm1[] = "dpLenV.data";
char frm2[] = "mseq.data";
char frm3[] = "z.data";
int temp;
int ecoun; /* number of error bits in each 6-bit CW */
int ecw_count; /* number of error codewords */
int ecw_bits_count; /* number of error bits in a codeword */
long int t_ecoun; /* total number of error symbols */
long int t_ecoun_coding; /* total number of error symbols after err
correction */
int badBlkV [symbols/BLEN]; /* damaged block vector */
int badBlkV_len = 0; /* damaged block vector length */
int badPacV [DPCOUNT_MAX+1]; /* damaged packet vector */
int badPacV_len = 0; /* damaged packet vector length */
int badPacVleft [symbols/BLEN]; /* left-index vector for badpacs */
int badPacVright [symbols/BLEN]; /* right-index vector for badpacs */
int badPacVleft_len = 0; /* length of left-index vector for bad-
pacs */
short Z [symbols]; /* detected symbol vector */
int Z_len = 0; /* detected symbol vector length */
short pV [symbols]; /* original symbol vector */
int dpv_len = 0; /* total bits in DL, len of dpv */
int dpLenV [DPCOUNT_MAX+1]; /* original LPDU vector */
short ErrBits[symbols]; /* bits in error */
int j, k, sumLen, left, right;
/* ----- load dpLenV, mseq(pV), and Z */
if ((fp = fopen(frm1, "rb"))==NULL)
printf("can't open dpLenV.data for reading\n");

```

```

while ( doneFlag != 1 )
{
get_dpLenV();
gmsk();
dl_assemble_frames();
doneFlag = (outs.t_badPacs>100 && outs.t_badBlks>100);
outs.Eb_ch = 10.0*log10(Eb_ch / (double)(outs.t_blks*BLEN));
} /* end of while */
Eb_ch /= (double)(outs.t_blks*BLEN);
} /* end of for */
)

```

dlGen.c

```

/*****
*/
/* dlGen.c
*/
/*****
#include "global.h"
long symbols, /* number of symbols */
L; /* number of samples, = symbols*s */
void get_dpLenV();
/*****
*/ Name: get_dpLenV()
/* Description: Generate Datalink layer random-length
*/ vector, and store into a file called
*/ "dpLenV.data"
/* Input: None
/* Output: None
/*****
void get_dpLenV()
{
int dpLenV [DPCOUNT_MAX+1]; /* +1 coz the first one is dummy "preLen" */
int preLen; /* length of dummy packet */
float rn; /* length of dpv */
preLen = (int)(INFLEN * (double)(rand()/(RAND_MAX+1.0)));
dpLenV[0] = preLen;
dpV_len = preLen;
for (int i=1; i<DPCOUNT_MAX+1; i++)
{
rn = (double)(rand()/(RAND_MAX+1.0));
if (rn >= 2.0/3.0)
dpLenV[i] = SLEN;
else
dpLenV[i] = (rn >= 1.0/3.0)? MLEN:LLEN;
dpV_len += dpLenV[i];
}
if ((fp = fopen("dpLenV.data", "wb"))==NULL)
printf("can't open dpLenV.data for writing\n");
else fwrite((void *) dpLenV, sizeof(int), DPCOUNT_MAX+1, fp);
fclose(fp);
/*----- Update global variables -----*/

```

```

else fread((void *) dpLenV, sizeof(int), DPCOUNT_MAX+1, fp);
fclose(fp);
if ((fp = fopen(fnm2, "rb"))==NULL)
    printf("can't open msecq.data for reading\n");
else fread((void *) epV, sizeof(short), symbols, fp);
fclose(fp);
/* ----- find length of Z */
if (dt_type == 'C')
    temp = (symbols%2 == 0)? 4 : 5; // number of symbols Z shorter than pv
else if (dt_type == 'D')
    temp = (symbols%2 == 0)? 4 : 5; // number of symbols Z shorter than pv
else
    printf("wrong detector type\n"); exit(1);
}
Z_len = symbols - temp;
if ((fp = fopen(fnm3, "rb"))==NULL)
    printf("can't open Z.data for reading\n");
else fread((void *) eZ, sizeof(short), Z_len, fp);
fclose(fp);
/* ----- complete sequence Z */
for (int i=0; i<temp; i++)
    Z[Z_len+i] = pv[Z_len+i];
Z_len = symbols;
/* ===== */
/* error processing */
/* ===== */
/* ----- finding the bad blocks in PHY layer */
t_count = 0;
t_count_coding = 0;
j = 0;
for (int i=0; i<symbols; i+=BLEN)
{
    ecw_count = 0;
    /* ----- travel in block by CW's */
    ecw_bits_count = 0;
    for (int blk_offset = 0; blk_offset < BLEN-1; blk_offset+=6)
    {
        ecw_count = 0;
        /* ----- travel in cw's */
        for (int bit_offset = 0; bit_offset < 6; bit_offset++)
            if (Z[i+blk_offset+bit_offset] != pv[i+blk_offset+bit_offset])
                ecw_count ++;
        if (ecw_count > 0)
        {
            ecw_count ++;
            t_count += ecw_count;
            ecw_bits_count += ecw_count;
        }
    }
    if (ecw_count > ERRMAX)
        badBlkV[j++] = (int)(i/BLEN);
    t_count_coding += ecw_bits_count;
}
}

)
badBlkV_len = j;
/* ----- finding the bad packets in DL layer based on badBlkV and
dpLenV */
j = 0;
dpV_len = 0;
for (int i=0; i<DPCOUNT_MAX+1; i++)
    dpV_len += dpLenV[i];
for (int i = 0; i < badBlkV_len; i++)
{
    left = badBlkV[i] * INFLEN; /* left bound of bad pacs in dpV */
    right = left + INFLEN - 1; /* right bound of bad pacs in dpV */
    k = 0; /* index of bad pacs in dpLenV */
    sumLen = dpLenV[0];
    while (left+1 > sumLen && sumLen < dpV_len) /* search left bound of bad
pacs */
    {
        k ++;
        sumLen += dpLenV[k];
    }
    badPacVleft[j] = k; /* store the left index */
    while (right+1 > sumLen && sumLen < dpV_len) /* search right bound of
badpacs */
    {
        k ++;
        sumLen += dpLenV[k];
    }
    badPacVright[j] = k; /* store the right index */
    j++;
}
badPacVleft_len = j;
/* ----- generate badPacV */
if (badPacVleft_len > 0)
{
    k = 1;
    badPacV[0] = badPacVleft[0];
    for (int i = 0; i < badPacVleft_len; i++)
    {
        for (int j = badPacVleft[i]; j <= badPacVright[i]; j++)
            if ( j != badPacV[k-1] )
                badPacV[k] = j;
                k ++;
    }
    badPacV_len = k;
}
else
    badPacV_len = 0;
/* ----- recording results - update outs */
outs.t_count += t_count;
outs.t_count_coding += t_count_coding;
outs.t_badInfBits_PHY += badBlkV_len * INFLEN;
}
}

```



```

/*****
/* Name: channel()
/* Description: Pass signal through either AWGN or Rayleigh fading channel
/* Input: I, Q - I&Q signal vectors
/* Output: ch_type: channel type
          idum: pointer to random number seed
          doLoad: flag to load pre rand # or not
          doSave: flag to save rand # or not
          None
/*****
void channel(
    double *I,
    double *Q,
    const char ch_type,
    const double fd,
    long **idum,
    int **doLoad,
    int **doSave
)
{
    double *ci = new double;
    double *cq = new double;
    double tmpi, tmpq;
    long int t, j;
    if (ch_type != 'A' && ch_type != 'F') {
        printf("Incorrect channel type");
        delete [] I; delete [] Q; exit(1);
    }
    if (ch_type == 'F') {
        for (t=0; t<L; t++)
        {
            gen_fades(ci, cq, (t+t_fade_init)*(double)(T)/(double)(s), fd);
            tmpi = I[t];
            tmpq = Q[t];
            I[t] = (*ci)*tmpi - (*cq)*tmpq;
            Q[t] = (*cq)*tmpi + (*ci)*tmpq;
            Eb_ch += (I[t]*I[t]+Q[t]*Q[t]);
        }
        *t_fade_init += t;
    }
    delete ci, cq;
    for(j=0; j < L-2; j++)
    {
        I[j] += randnorm(idum[0],doLoad[0],doSave[0]);
        Q[j] += randnorm2(idum[1],doLoad[1],doSave[1]);
    }
    *doSave[0]=1; *doSave[1]=1;
    I[j] += randnorm(idum[0],doLoad[0],doSave[0]);
    Q[j] += randnorm2(idum[1],doLoad[1],doSave[1]);
    j++;
    I[j] += randnorm(idum[0],doLoad[0],doSave[0]);
    Q[j] += randnorm2(idum[1],doLoad[1],doSave[1]);
}
/*****/

/*****
/* Name: gen_fades()
/* Description: generate fading signals
/* Input: ci, cq - I&Q fading signals
          t: time value
          fd: Doppler shift
/* Output: None
/*****
void gen_fades(double* ci, double* cq, double t, double fd)
{
    int NO = 32;
    double N = (2*NO+1)*2,
           fm = fd/19200.0/T,
           wm = 2 * PI * fm,
           A=0.0, B=0.0,
           w,
           beta;
    for (int n=1; n<=NO; n++)
    {
        w = wm * cos(2*PI*(double)(n)/N);
        beta = PI * (double)(n) / (double)(NO+1);
        A += cos(beta) * cos(w*t);
        B += sin(beta) * cos(w*t);
    }
    *ci = 2*A + sqrt(2)*cos(wm*t);
    *cq = 2*B;
}
/*****/

/*****
/* Name: conv_d()
/* Description: convolution of vector x (double) and h *
/* Input: x,h - input vectors, lenX=lenH
/* Output: y - result
/*****
void conv_d(double* x, int lenX, double* h, int lenH, double* y);
void conv(short* a, int lenA, double* b, int lenB, double* y);
/*****/

/*****
/* Name: filter.c
/* Description:
/*****
#include "global.h"
void conv_d(double* x, int lenX, double* h, int lenH, double* y);
void conv(short* a, int lenA, double* b, int lenB, double* y);
/*****/

/*****
/* Name: conv_d()
/* Description: convolution of vector x (double) and h *
/* Input: x,h - input vectors, lenX=lenH
/* Output: y - result
/*****
void conv_d(double* x, int lenX, double* h, int lenH, double* y)
{
    int i,k;
    if (lenH > lenX)
    {
        printf("Error, lenX should be no shorter than lenH");
        return;
    }
}
/*****/

```

```

)
for (i=0;i<lenH;i++)
(
    y[i]=0;
    for (k=i;k>=0;k--)
        Y[i] = Y[i] + x[i-k]*h[k];
)
for (i=lenH;i<lenX;i++)
(
    y[i]=0;
    for (k=lenH-1;k>=0;k--)
        Y[i] = Y[i] + x[i-k]*h[k];
)
return;
)
)
/*****
** Name: conv()
** Description: convolution of vector x (short) and h */
** Input: a,b - input vectors, lenA>=lenB
** Output: y - result
**/
void conv(short* a, int lenA, double* b, int lenB, double* y)
(
    long int i,k;
    if (lenB > lenA)
    (
        printf("Error, lenB > lenA");
        return;
    )
    for (i=0;i<lenB;i++)
    (
        y[i]=0;
        for (k=i;k>=0;k--)
            Y[i] = Y[i] + a[i-k]*b[k];
    )
    for (i=lenB;i<lenA;i++)
    (
        y[i]=0;
        for (k=lenB-1;k>=0;k--)
            Y[i] = Y[i] + a[i-k]*b[k];
    )
    return;
)
)
}

double ran2(long *idum, int *doLoad, int *doSave);
double randnorm2(long *idum, int *doLoad, int *doSave);
/* Generate gaussian random double
* - if *doLoad=1 (this run is not the first-run, needs to load previous
* noise table, to keep noise continuity), *doLoad flag is turned on in
* main function, turned off after each load
* - if *doSave=1 (this run is the last run, needs to save current noise
* parameter table into file "NoiseTable.data"), *doSave is turned on in
* channel.c, turned off after each save. Set *doSave twice ensure the
* flag is passed along to ran1.c
* - randnorm() and randnorm2() can generate two independent Gaussian rand
* number, given two largely separated feeds.
*/
double ran1(long *idum, int *doLoad, int *doSave)
(
    int j;
    long k;
    static long iy=0;
    static long iv[NTAB];
    double temp;
    FILE *fp;
    static struct (long s_idum; long s_iy; long s_iv[NTAB]); nTable;
    if (*doLoad == 1)
    (
        if ((fp=fopen("NoiseTable1.data", "rb")) == NULL){
            printf("can't open NoiseTable1.data for reading\n"); exit(1);
        }
        else fread((void *) &nTable, sizeof(nTable), 1, fp);
        fclose(fp);
        *idum = nTable.s_idum;
        iy = nTable.s_iy;
        for (int i=0; i<NTAB; i++)
            iv[i] = nTable.s_iv[i];
        *doLoad = 0;
    )
    if (*idum <= 0 || !iy) (
        if (-(*idum) < 1) *idum=1;
        else *idum = -(*idum);
        for (j=NTAB+7;j>=0;j--) (
            k=(*idum)/IQ;
            *idum=IA*( *idum-k*IQ)-IR*k;
            if (*idum < 0) *idum += IM;
            if (j < NTAB) iv[j] = *idum;
        )
        iy=iv[0];
    )
    k=(*idum)/IQ;
    *idum=IA*( *idum-k*IQ)-IR*k;
    if (*idum < 0) *idum += IM;
    j=iy/NDIV;
    iy=iv[j];
    iv[j] = *idum;
    if (*doSave == 1)
    (
        nTable.s_idum = *idum;
        nTable.s_iy = iy;
        for (int i=0; i<NTAB; i++)

```

randnorm.c

```

/*****
**
** randnorm.c
**/
#include "global.h"
double ran1(long *idum, int *doLoad, int *doSave);
double randnorm(long *idum, int *doLoad, int *doSave);

```

