MULTIPLE ACCESS AND CODING METHOD FOR WIRELESS ATM

BY

CHENG SIU LUNG

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

DIVISION OF INFORMATION ENGINEERING

THE CHINESE UNIVERSITY OF HONG KONG

JULY 1999

摘要

預測排隊複用(PQMA, Predictive Queuing Multiple Access) 是一個媒體可用協議, 用來運輸聲音, 資料和視像. 主要是在室內的無線信道上運作. 我們修改異步傳輸模式(ATM)的 5 個位元組幀頭, 使其可以做到多速率數據包協議, 而且有自動重發請求的能力. 我們用 VCI 達到彈性帶寬分配的效果. 在 ATM 的幀頭裡面. 有一個預測數值, 基站用這個數值去安排各移動站的輸送資料時間. 在衰減的信道, 這個協議有自動重發的能力. 我們用慢頻跳動的方法去應付頻率選擇性的衰減. 我們分析及模擬了 PQMA 的排隊性和錯誤表現而顯示出 PQMA 可以用在個人通訊系統上的多媒體通訊.

快速碼(Turbo Code)在一個高斯白噪聲信道上有很理想的表現,所以 PQMA 會利用快速碼去減低傳輸上的錯誤.爲了減少在慢平瑞利衰減(Slow flat Rayleigh fading channel)的環境下傳輸的錯誤, 本文將會提出一種利用快速碼特性的新技術去估計所傳輸的信道的特性並將估計所得出的資料去增加解碼時的準確性.透過電腦在不同環境下的模擬, 我們發覺新技術可以大大提高解碼時的準確性, 甚至可以接近理論上所可以取得的極限.

Acknowledgement

I would like to express my deepest gratitude to my supervisor Prof. Joseph Y. Hui for his critical evaluation of ideas, which acts as a valuable guidance and advice for my research work in the past three years.

I thank also Prof. Victor Wei, who is so generous in sharing his knowledge in coding theories with me. I am please to have discussions with my friends Terrence, Eric and Tze Lung. In addition, I would like to thank my friends Edward for keeping our laboratory's equipments in good condition and provide an excellent environment for research works.

Abbreviation

ABR: Available Bit Rate

ARQ: Automatic repeat request

ATM: Asynchronous Transfer Mode

AWGN: Addictive White Gaussian Noise

BER: Bit Error Rate

CBR: Constant Bit Rate

CLP: Cell Lost Priority

FER: Frame Error Rate

FIR: Finite Impulse Response

HEC: Header Error Check

GSM: Global System for Mobile Communications

LLR: Logarithm of Likelihood Ratio

nrt_VBR: non-real-time Variable Bit Rate

PCS: Personal Communication Systems

PQMA: Predictive Queuing Multiple Access

PTI: Payload Type Identifier

QoS: Quality of Service

RCPTC: Rate Compatible Punctured Turbo Code

rt_VBR: real-time Variable Bit Rate

SNR: Signal to Noise Ratio

VCI: Virtual Circuit Identifier

VPI: Virtual Path Identifier

UBR: Unspecified Bit Rate

ABSTRACT

PQMA (Predictive Queuing Multiple Access) is a media access protocol for transporting speech, data, and video via indoor wireless channels. The 5-byte ATM header is modified to accommodate a multi-rate packet access protocol with ARQ capability. Similar to ATM, flexible bandwidth assignment is achieved by the use of VCI. The ATM header contains a prediction field for the base station to schedule subsequent transmission from the mobile. Packet queues are maintained at both the base station and the mobile station. The protocol also has a built in capability for retransmission required for fading channels. Slow Frequency Hopping is also employed for combating frequency selective fading. The queueing and error performance of PQMA is analyzed and simulated, which demonstrates that PQMA can support multimedia communication for PCS.

Turbo coding has shown impressive performance in an AWGN channel. PQMA adopts turbo codes for error control. We propose an Iterative Channel Estimation technique for turbo codes over slow frequency hopped multiple access in a slow flat Rayleigh fading environment. Iterative Channel Estimation uses the intermediated decoder output of each

turbo decoding iteration to estimate the channel state information. Through simulation, Iterative Channel Estimation has demonstrated a performance very close to that of using perfect channel state information. We have simulated the performance of applying turbo codes with a slow frequency hopping scheme over slow flat Rayleigh fading channels. Our results show a performance close to the theoretical limit of a Rayleigh channel.

Contents

ABBREVIATION **3**

INTRODUCTION **9**

1.1 Wireless ATM for multimedia application 9

1.2 Challenges in Wireless ATM 11

1.3 Outline of thesis 14

PREDICTIVE QUEUING MULTIPLE ACCESS **17**

2.1 Introduction 17

2.2 Protocol for Mobile to Base 20

2.3 Scheduling Protocol at the Base Station 23

2.4 Rate Compatible Punctured Turbo code 27

2.5 FEC and ARQ methodologies 30

2.6 Experimental Results 32

2.7 Conclusion 34

FUNDAMENTALS OF THE WIRELESS COMMUNICATION MEDIUM **36**

3.1 Introduction 36

3.2 Error control and channel capacity 38

3.3 Capacity of fading channel 39

ITERATIVE CHANNEL ESTIMATION FOR TURBO CODE FOR FREQUENCY HOPPED MULTIPLE ACCESSING **45**

4.1 INTRODUCTION 45

4.2. TURBO CODE STRUCTURES 47

4.3 SYSTEM MODEL 49

4.4 ITERATIVE CHANNEL ESTIMATOR 53

4.5 TURBO DECODING WITH ITERATIVE CHANNEL ESTIMATION 56

4.6 SIMULATION RESULTS 58

4.7 CONCLUSION 63

DUMMY BITS INSERTED TURBO CODE **64**

5.1 INTRODUCTION 64

5.2 WEIGHT DISTRIBUTION OF TURBO CODES 66

5.3 ENCODING WITH DUMMY BIT INSERTION 69

5.3.1 Dummy bit insertion methodology *69*

5.3.2 Hybrid Periodic Random Interleaver *70*

5.3.3 Dummy bit removal before transmission *71*

5.4 DECODING WITH DUMMY SIGNAL ENHANCEMENT 73

5.5 WEIGHT DISTRIBUTION OF DUMMY BIT INSERTED TURBO CODING 76

5.6 SIMULATION RESULTS 77

5.7 SUMMARY 79

REFERENCES **81**

Chapter 1

Introduction

1.1 Wireless ATM for multimedia application

Modern wireless communication is evolving to an era of multimedia. We can see more wireless multimedia applications such as mobile telephony, wireless internet, teleconferencing and portable video on demand. Existing wireless communication systems such as GSM (Global System for Mobile Communications) and PCS (Personal Communication Systems) are not capable of supporting the variety of multimedia applications due to the diverse bandwidth requirements for the multimedia services.

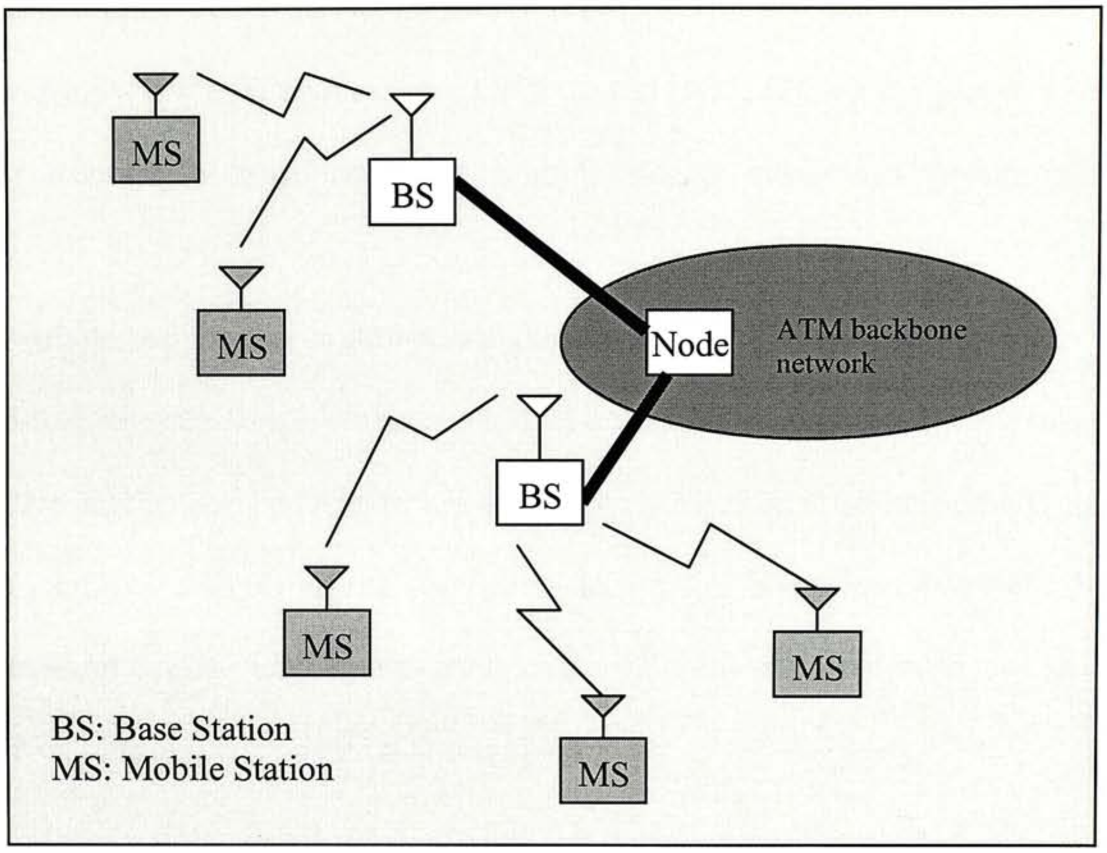

Fig. 1.1, Wireless / Wired ATM network

In a wired system, B-ISDN (Broadband Integrated Service Digital Network) makes the distinction between synchronous services (voice and video) and asynchronous services (data). An ATM system, however, is even more sophisticated in supporting the multimedia applications. ATM-based systems have categorized multimedia applications in five categories: (a) constant bit rate (CBR), (b) real-time variable bit rate (rt_VBR), (c) non-real-time variable bit rate (nrt_VBR), (d) unspecified bit rate (UBR) and (e) available bit rate (ABR) services [Toh97]. ATM systems provide flexible bandwidth on demand and allow quality of service (QoS) control. With fast Virtual Circuit Identifier (VCI) switching for broadband network, an ATM-based system is ideal for multimedia.

The evolution of wireless multimedia applications and ATM systems introduces an interesting research area – Wireless ATM. Wireless ATM combines the advantage of mobility and the QoS control for wired ATM systems. With wireless ATM, we are able to access multimedia applications anywhere and anytime. However, due to the difference between wired and wireless media, problems such as the design of media access protocol and error control scheme remain to be solved.

1.2 Challenges in Wireless ATM

Fundamental differences between wired and wireless media present great challenges in designing a wireless ATM system. Such differences have also raised doubts in the feasibility of building a wireless ATM system.

ATM was designed for a wired media with very low bit error rates (BER) of about 10^{-10}. A wireless medium, however, is very noisy and time varying. An ATM system does not provide a low level error correction scheme. Error correction is done at the transport protocol level. In wireless ATM, it is a great challenge for real-time services that require high bandwidth and fidelity. Noisy channels also cause much difficulty in the control of the system. Wireless ATM provides flexible on-demand bandwidth allocation, making wireless ATM distinguishable from the existing wireless systems such as GSM and PCS which use primarily circuit switching. Besides the wireless medium being corrupted by noise, flexible bandwidth allocation is not possible unless the mobile stations (MS) can effectively access the channel. Media access protocol is therefore a key topic in the design of a wireless ATM system.

In a wired ATM system, a multiplexer serves the media access function. In a wireless ATM system, mobile users are distributed geographically, all sharing the same noisy wireless media. The challenge for media access protocol is how to have distributed contention and access of ATM slots, and dynamically assigning bandwidth to the mobile users.

Also, wireless media is limited in bandwidth. Wired ATM often uses optics as the transmission medium, which has a bandwidth in the order of gigabits per second. The wireless media, however, has a bandwidth limited to about 34 Mb/s [KM97]. There is a basic question whether such limited bandwidth is capable of supporting multimedia services. A typical video stream has a bandwidth ranging from 1 to 1.5 Mb/s. The wireless media may easily be overloaded.

Moreover, as ATM was designed for riched bandwidth, the ATM cell header is big as compare to its payload. Such a big header is for the trade off for simplicity in switching. An ATM cell header uses up about 10 percent of the whole cell payload, which is too expensive in a wireless communication system. As shown in Fig. 1.2, many researchers have proposed wireless systems with media access protocol piggy backed to the existing ATM cell.

Wireless header	Wired ATM header	Cell Payload

Fig. 1.2, Wireless ATM cell with piggy backed wireless header.

As the existing ATM cell header is already considered too long for wireless ATM, any additional overhead is not justified for wireless ATM. As shown in Fig. 1.3, we propose a wireless ATM protocol called Predictive Queuing Multiple Access (PQMA) [FYP96], with a modified ATM cell header. PQMA is capable of multiple access control in handling the variety of services as proposed for ATM systems. A base station (BS) takes up the role of translating the cell header between wired and wireless media so that PQMA can be seamlessly integrated with wired ATM system.

Modified ATM header	Cell Payload

Fig. 1.3, Wireless ATM cell with modified ATM cell header.

1.3 Outline of thesis

In chapter 2, we shall describe our proposed wireless ATM multiple access protocol PQMA. It has been noted that the turbo code proposed by C. BERROU, A. GLAVIEUX and P. Thitimajshima [BGT93] has strong error correction capability in low Signal to Noise (SNR) environment. We have integrated PQMA with Rate Compatible Punctured

Turbo Code (RCPTC) [JPDB97]. For different categories of services, we adopt turbo code with different puncturing scheme. With different levels of puncturing, we can adaptively change the code rate according to the channel condition.

Chapter 3 focuses on the fundamental of wireless communication media and discusses the theoretical limits on the channel capacities of two common channel model, namely the Gaussian channel and the Rayleigh fading channel. Such theoretical limits are fundamentals to the evaluation of our proposed coding schemes. We shall also see under what condition can such limits be possibly achieved.

The Rayleigh fading channel is commonly used to model a wireless communication channel. Though turbo coding has shown impressive performance for Gaussian channel, it has been shown that turbo coding fails for slow flat Rayleigh fading channel. In chapter 4, we propose to apply turbo coding over a frequency hopped channel with channel interleaving so as to combat the Rayleigh fading. We propose a new channel estimation technique called Iterative Channel Estimation to estimate the channel state information. Through simulation, we have shown Iterative Channel Estimation can converge to a very accurate estimate for the channel state information.

In chapter 5, we draw our attention to turbo coding with small frame size. Speech communication systems are generally characterized with small frame size. It is been noted that turbo coding degrades when the information block size decreases. We propose a dummy bit inserted turbo code aiming to improve the performance of turbo code over small frames. Inserting dummy bits in the information stream can reduce the multiplicity of the low-weigh codewords. However, since dummy bits introduce extra parity bits, it is shown that such extra redundancy is not justified as the performance gain is not significant.

Chapter 2

Predictive Queuing Multiple Access

2.1 Introduction

New applications are developing for wireless communications based on portable computers. Such terminals can provide functions such as mobile telephony, electronic mail, and file transfer. Higher bandwidth applications such as Web browsing, multimedia presentation, video on demand, and video conferencing are becoming popular on notebook computers.

There is a need to use the wireless channel for broadband access to these computers. Such effort is broadly termed wireless multimedia.

For wireline networks ATM was devised to allow flexible and on-demand allocation of transmission capacity for broadband networks. Also, ATM allows fast switching using Virtual Circuit Identifiers (VCI) and parallel switching fabrics. More importantly, Quality of Service (QoS) can be carefully provisioned for different media types with vastly different burstiness, bit-rate, delay and error requirements. The ATM cell header is shown in fig 2.1.

For wireless networks supporting high bandwidth multimedia applications, it is desirable to maintain compatibility with the wireline ATM network, using similar notions of fixed cell size, VCI routing, and QoS provisioning. The advantages of ATM apply similarly for the wireless environment. However, the ATM header and protocol functions must be modified to reflect major differences between the wireless and wireline networks.

<table>
<tr><td colspan="3">VPI</td></tr>
<tr><td>VPI</td><td colspan="2">VCI</td></tr>
<tr><td colspan="3">VCI</td></tr>
<tr><td>VCI</td><td>PLT</td><td>CLP</td></tr>
<tr><td colspan="3">HEC</td></tr>
</table>

VPI – Virtual Path ID

VCI – Virtual Channel ID

PLT – Playload Type

Fig. 2.1 Wireline ATM Cell Header

The first difference is that mobiles require the use of scheduling or contention algorithms in order to access the wireless channel. Such algorithms allow bursty and multi-rate establishment of multimedia virtual circuits.

The second difference is that a severely impaired wireless transmission environment requires the extensive use of both forward error correction coding and ARQ. Sporadic errors as well as changing channel condition may cause fluctuation in the required channel bit rate due to code adaptation and retransmission.

The third difference is that bandwidth is severely limited on the wireless channel. Multiple frequency bands may be necessary for high bandwidth applications.

We propose to modify the forward and reverse channel ATM header format to accommodate the different functionalities for a wireless channel [FYP96]. These functionalities include scheduling channel access and error control, both FEC and ARQ. Slow frequency hopping is adopted for providing more bandwidth and diversity protection against fading.

2.2 Protocol for Mobile to Base

We assume that ATM cells queue at both the base and the mobile for transmission. VPI/VCI translation occurs prior to the cell being put into the queue at the base station. The number of enqueued cells at either the base station or the mobile depends on the data generation rate of the connection, retransmission requests resulting from channel errors, and the rate at which cells are scheduled for transmission. Queue length could be one factor for scheduling transmission for connections in a microcell.

Each connection on the wireless ATM channel is labeled by 1 byte (fig 2.2), instead of the usual 3 1/2 byte VPI/VCI field for wireline ATM. Therefore as many as 256 virtual channels can be supported in a microcell. This should be sufficient since a microcell may have limited bandwidth.

<table>
<tr><td colspan="4">VCI</td></tr>
<tr><td colspan="4">PF</td></tr>
<tr><td colspan="2">PF</td><td colspan="2">SEQ</td></tr>
<tr><td>Code Rate</td><td>New</td><td>PTI</td><td>CLP</td></tr>
<tr><td colspan="4">HEC</td></tr>
</table>

PF – Prediction Field

SEQ – Sequence No.

Fig. 2.2 Wireless ATM Cell Header

The key concept for PQMA is the use of a prediction field in the cell header for explicit scheduling of the next cell for mobile to base communication. For each virtual circuit, a currently transmitted cell schedules the next cell by the parameter Prediction Field (PF), which represents the offset in term of slots from the current cell (fig 2.3). The offset depends on the current bit-rate of the circuit, and is approximately given by the total channel bit-rate divided by the bit-rate of the virtual circuit after channel coding.

This offset may fluctuate from cell to cell due to bit-rate changes for a connection, retransmission needs, and fill status of the source buffer. The offset could be reduced if the source buffer is close to full, or increased if the source buffer is close to empty. Retransmissions tend to increase buffer occupancy and therefore reduces the offset.

This offset is coded as two sub-fields in the Prediction Field as frame offset and slot offset. The wireless channel is divided into frames. Each frame contains 32 slots. Each slot may carry a 53 byte cell. The 7-bit frame offset refers to the frame location of the next schedule cell. The 5-bit slot offset refers to the slot location in the referenced frame.

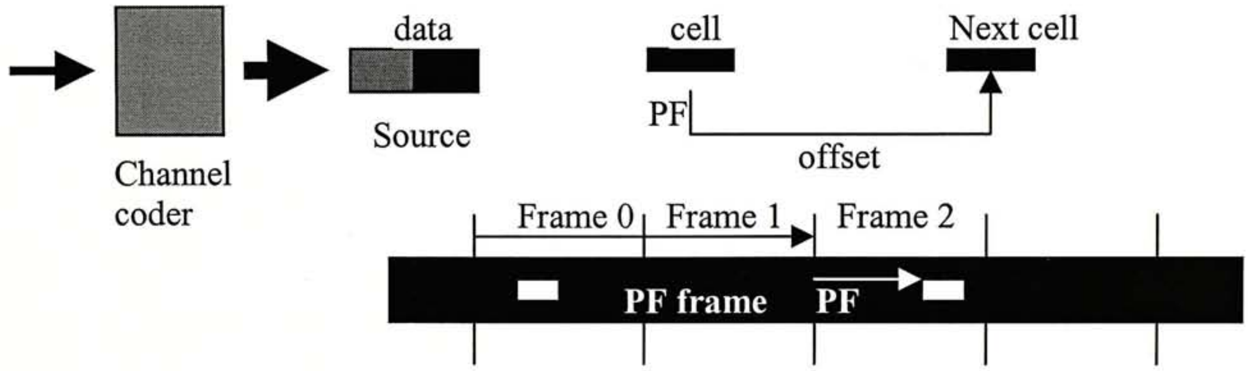

Fig. 2.3 Offset and its Representation

One byte of the 5-byte ATM header is used for error control functions. First, a 4-bit sequence number (SEQ) represents cyclically the cell sequence number. This is used for selective repeat of errored transmission. A 3-bit code rate field represents the error control code being used. A 1-bit New field is set to 1 if the cell is the first cell of a connection. We shall describe in detail the error control mechanism in a later section.

The remaining 1 1/2 byte is identical to that for wireline ATM, namely a 3-bit Payload Type Identifier (PTI), a 1-bit Cell Loss Priority (CLP), and an 8-bit Header Error Check (HEC) field.

2.3 Scheduling Protocol at the Base Station

Scheduling is performed primarily at the base station. A schedule table (fig 2.4) is kept, which records the VCI scheduled for slots sequentially. Besides the VCI, the slot entry in the table also records the PTI of the VCI and a delay value to be explained later. A pointer PTR is used to reference the current frame in the table.

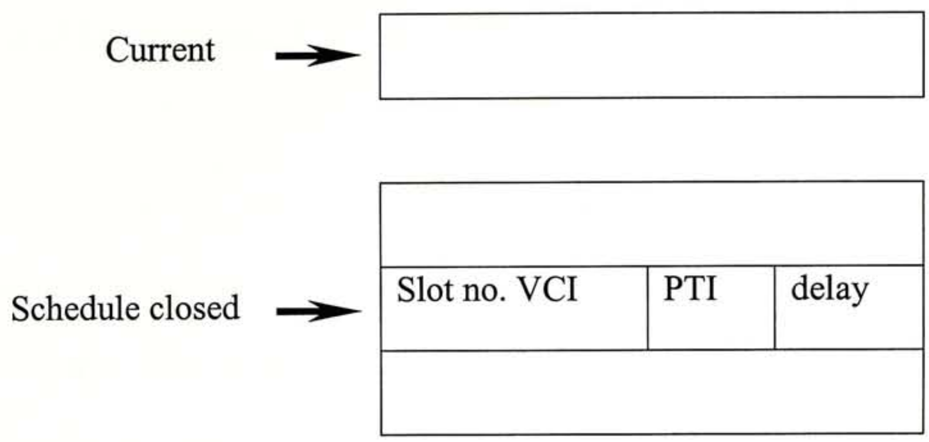

Fig. 2.4 Schedule Table at Base

When an ATM cell arrives, the base examines the entry in the table indexed by PTR plus the offset indicated by the PF field. If the slot is not scheduled, then the VCI, the PTI, and delay=0 is entered in the empty entry. If the entry is occupied, then a contention arises. The contention is resolved first by using the PTI field. If the PTI happens to be identical, then the contending party with a larger delay wins. The losing party then contends for the next slot with delay increased by one.

At the beginning of each frame, the base broadcast the frame schedule for the next (PTR+1) frame. This broadcast takes up one slot in frame PTR, listing sequentially the

VCI entries for frame PTR+1 in the schedule table. For a frame size of 32 slots, 32 bytes are used for indicating the VCIs.

Since the scheduling is explicit, the base expects the mobile to send a cell in the scheduled slot. If for some reasons the reception fails, the base would know immediately and request a retransmission. The requests could be made in the next frame using the broadcast slot. The negative acknowledgements (NAKs) are represented by a 1 byte VCI and a 4-bit SEQ. The mobile then selectively repeats the cell being negatively acknowledged. Excessive number of NAKs for a VCI may triggers an exception control condition, which indicates either a channel or protocol failure.

Two additional variations of this basic scheduling algorithm could be made. First, the unscheduled slots could be used by mobiles using random access methods such as Aloha. The unscheduled slots could be used for selective repeat or for easing temporary backlog in the buffer of a mobile. If the reception at the base is successful, positive acknowledgement (ACKS) could be sent in the broadcast slot in the next frame, indicating the VCI and SEQ of the successful contender.

Second, scheduling can also be performed in the frequency domain besides the time domain. This is particularly useful if slow frequency hopping is employed. The broadcast slot now contains the list of VCIs for all slots referenced by both time and frequency within a frame duration.

Besides the broadcast slot, protocol information is also carried by the ATM headers of the forward (base to mobile) channels. For base to mobile communications, the base station can easily schedule slots for virtual circuits. The ATM header contains a 1-byte VCI field. The mobile examines the VCI field to see if the ATM cell is marked for the mobile. Instead of inband transport of the VCI within the ATM cell, out-of-band transport of the VCI in the broadcast slot can also be used. The 1 1/2 byte PF field is not necessary for the forward channel. The rest of the ATM header remains the same as the reverse channel.

2.4 Rate Compatible Punctured Turbo code

The classical turbo code shown in Fig. 4.1 is of rate 1/2. Without puncturing, a rate 1/3 is obtained. The classical turbo code has shown a very impressive performance and is capable of achieving a BER of 10^{-5} in low SNR. In wireless ATM, services of different QoS have diverse requirements on the BER. A BER of 10^{-5} is too strict for some services such as speech and video, which have BER requirements ranging from 10^{-3} to 10^{-4}. On the other hand, the fading nature of wireless channel may cause a fluctuation in the channel state. To fully utilize the valuable channel capacity of the wireless channel, a variable rate coding scheme is generally applied to provide BER performance that matches the QoS requirement of a particular service under a particular channel state.

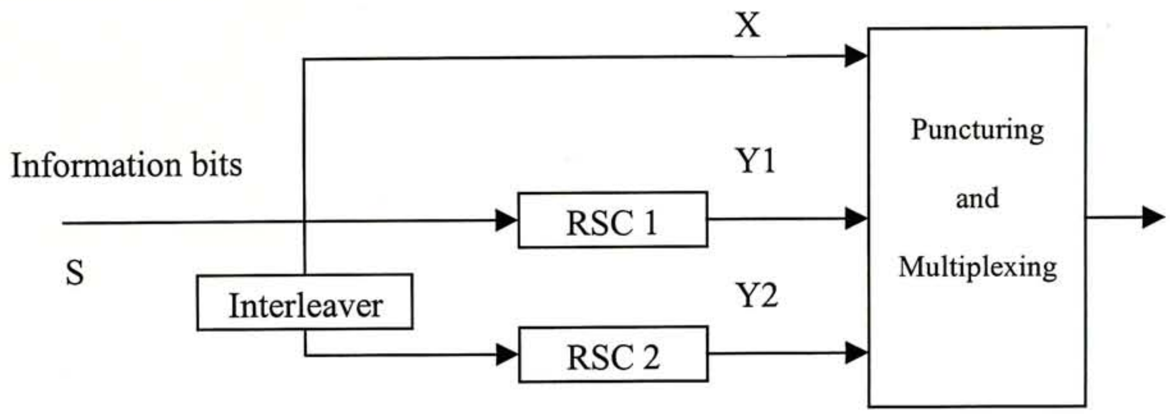

Fig. 2.5 Rate Compatible Puncture Turbo codes encoder

P. Jung and J. Plechinger [JPDB97] proposed the Rate Compatible Puncture Turbo codes, RCPTC's, which is able to provide variable rate turbo codes that match different BER requirements. As shown in Fig. 2.5, RCPTC consists of two component RSC encoders and a puncturing and multiplexing device. Depending on the puncturing schemes, the minimum code rate is 1/3 and the maximum code rate is 1. Turbo coder can iteratively decode the received stream.

Different puncturing schemes result in different performance. Berrou's puncturing proposed by C. BERROU, A. GLAVIEUX and P. Thitimajshima [BGT93] only punctures the parity bits but not the information bits. P.Jung and J. Plechinger [JPDB97] propose UKL puncturing which partially punctures both the information bits and the parity bits. The BER performances of different puncturing schemes are shown in Fig. 2.6. It is noted that Berrou's puncturing facilitates a better performance at lower values of E_b/N_0 whereas ULK puncturing outperforms in higher E_b/N_0 value. The cross over point is at BER 10^{-4} and moves to lower BER with increasing code rate.

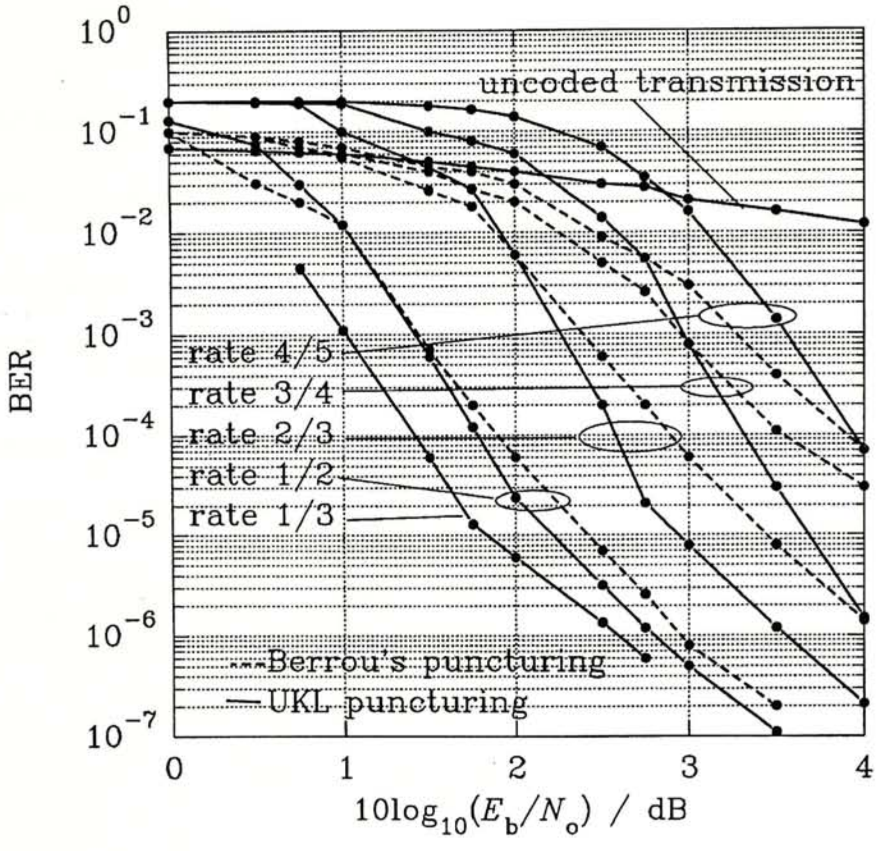

Fig. 2.6, Performance of an RCPTC in terms of BER versus E_b/N_0; block size 672 bit, 10 decoding iterations, AWGN channel [WROL99, pp 252].

2.5 FEC and ARQ methodologies

In this section, we examine how errors may arise and be dealt with. Errors may affect the functioning of the protocol described, or corrupt the data transport directly.

Protocol information errors can arise if the broadcast slot, the VCI, PF, or NAK are corrupted. Since scheduling is explicit, the base station could detect such errors if the mobile fails to send the scheduled cell. NAKs are sent immediately under such conditions. For lost PFs, the base can use previous values of PF for the virtual circuit, since the value of PF need not be exact. We believe that our scheme is robust to occasional protocol information errors of various types. If such errors are persistent, the base triggers an exception condition.

Several methods are available for handling corrupted data transport. As mentioned earlier, the 48-byte information payload may contain a CRC itself for detecting errors within the cell. A NAK could be sent requesting a retransmission if an error is detected.

Speech service have a BER requirement range from 2.5×10^{-2} to 1×10^{-3} which favors the choice of Berrou's puncturing. For video and data service, BER is required to be less than 10^{-4}. Therefore, the ULK puncturing is more suitable. In table 2.1, the assignment of the 3-bit code rate field in PQMA is shown with different puncturing scheme for different services.

Table 2.1 Assignment of the code rate field in PQMA for speech service.

Code rate field in PQMA	Puncturing scheme	Code rate of RCPTC
000	Berrou's	3/4
001	Berrou's	2/3
010	Berrou's	1/2
011	Berrou's	1/3
100	ULK	3/4
101	ULK	2/3
110	ULK	1/2
111	ULK	1/3

We propose this wireless ATM scheme to be used in conjunction with Slow Frequency Hopping (SFH) with moderate interleaving for the purpose of combating frequency selective fading. We have also explored the possibilities of an adaptive transmission baud-rate, which may allow a higher transmission rate if a direct line of sight propagation condition exists.

2.6 Experimental Results

Extensive simulation is performed for the following system. For this thesis, we use the following example. We assume for indoor radio that a total of 20 Mb/s capacity is available. Slow Frequency hopping with rate 1/2 convolutional coding and interleaving are employed. We assume three kinds of services, namely 16 Kb/s speech, 1.5 Mb/s video, and Web data at 1 Mb/s. These bit rates are expanded by 2 upon convolutionally coded. An active terminal use any one of the three services at one time with probability 0.4, 0.2, 0.4 respectively. Call admission control can be determined by examining the region where delay for these services is acceptable. For this example, we examine throughput and delay versus the number of active terminals with the above service type probabilities.

Figure 2.7 illustrates the throughput as more active terminals are admitted. The straight line shown is the expected load of the system. We can see that the throughput saturates at around 10 terminals. PQMA adopts earliest deadline first scheme to resolve the contention of cells having identical PTI. The queuing delay suffered by video according to the PQMA scheme with earliest deadline first scheme is shown in figure 2.8.

The above simulation is performed under the assumption that the channel is error free. In order to see under what regime we may assume an error free condition, we have simulated in chapter 4 the performance of turbo code applied over slow flat rayleigh channel with channel interleaving and slow frequency hopping. It has been shown that when the number of channels and message length increase, we can assume the channel is error free if the SNR is increased to 4.5 dB. With RCPTC, it is possible to adjust the code rate adaptively according to the channel condition of a particular frequency band.

2.7 Conclusion

We propose a wireless ATM protocol which provides true on demand bandwidth allocation using the notion of timed reservation via PQMA. Preliminary results show that throughput and delay are acceptable for indoor applications.

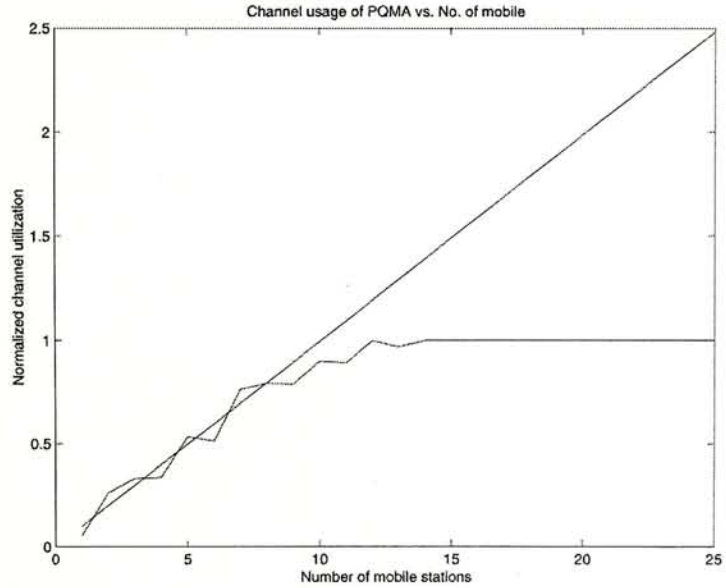

Fig. 2.7 Throughput for PQMA

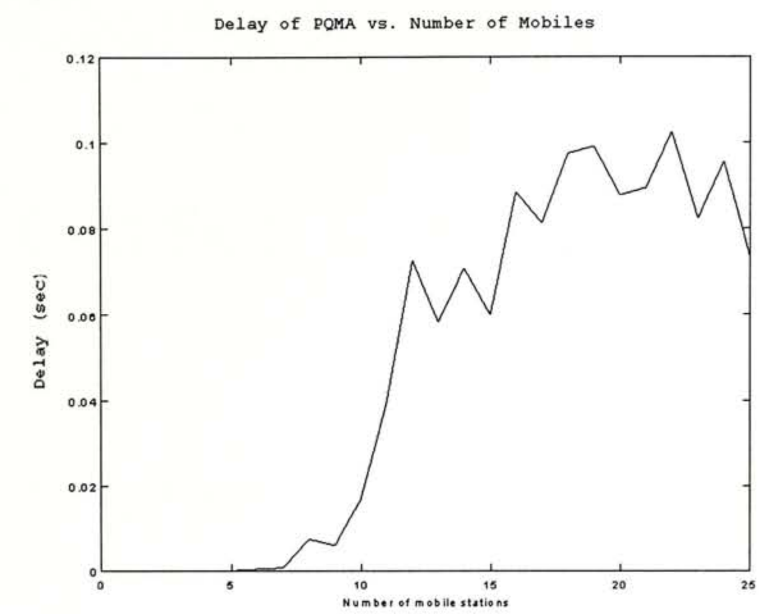

Fig. 2.8 Delay for PQMA

Chapter 3

Fundamentals of the Wireless Communication Medium

3.1 Introduction

One of the most challenging topics in a wireless ATM system is the error control problem. PQMA adopts turbo code for error correction control. In this chapter, we shall focus on the theoretical limits of the channel capacity of an AWGN channel and a Rayleigh fading channel. C. BERROU, A. GLAVIEUX and P. Thitimajshima [BGT93] have shown that turbo codes can achieve near channel capacity performance in AWGN channels. While slow flat Rayleigh fading channels are commonly used to model wireless communication channels, we shall focus on studying the performance of turbo codes over such channels.

A Gaussian channel is a commonly used channel model in wireless communication studies. Much research in evaluating the performance of a wireless system or coding system is based on the Gaussian channel model.

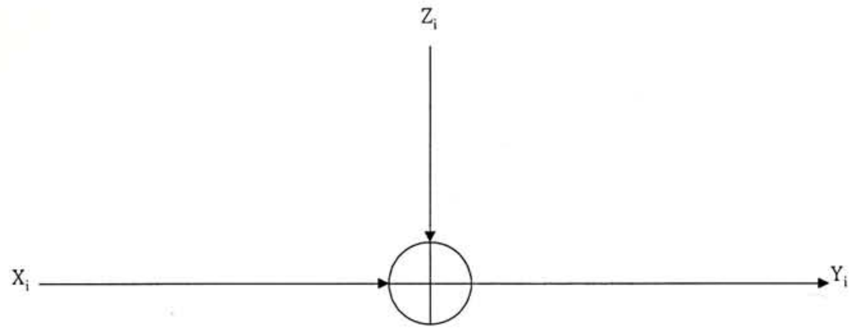

Fig. 3.1, The Gaussian channel

Fig. 3.1 shows a time discrete channel with output Y_i at time i, where Y_i is the sum of the input X_i and the noise Z_i. Z_i is the Gaussian noise with a Gaussian Distribution of variance N. Thus,

$$Y_i = X_i + Z_i \qquad\qquad Z_i \sim N(0, N) \tag{3.1}$$

Z_i is independent of X_i. The common limitation on the input is an energy or power constraint where

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 \le P \tag{3.2}$$

for any codeword $(x_1, x_2, \ldots, x_n)$ transmitted over the channel.

3.2 Error control and channel capacity

In 1948, Shannon [Sha48] derived the capacity of a AWGN channel as

$$C = W \log(1 + \frac{E_s}{N_0}) \quad \text{bits per second} \tag{3.3}$$

Where W is the bandwidth of the channel and E_s is the average signal energy. N_0 is the power spectral density (PSD) of the Gaussian noise. Shannon has also shown that as long as the information transmission rate is lower than the capacity, there exists an error correction code that can provide a high level of reliability at the receiver output. Equation 3.3 can be used to find the limit on the coding gain of a system. Suppose we use a BPSK-modulated system. BPSK-modulation is good for evaluating a coding scheme because it has a spectral efficiency of roughly1. Spectral efficiency is the number of bits that can be sent per two-dimensional signaling interval of duration T. Let η be the spectral efficiency expressed in terms of bits per second per Hertz (b/s/Hz). Now, we have $E_s/N_0 = \eta E_b/N_0$ and $\eta = C/W$, then

$$\frac{C}{W} < \log(1+\frac{E_s}{N_0}) \tag{3.4}$$

$$2^{\eta} < 1+\eta\frac{E_b}{N_0} \tag{3.5}$$

$$\frac{E_b}{N_0} > \frac{2^{\eta}-1}{\eta} \tag{3.6}$$

From equation 3.6, we can easily see that the limit in the SNR for a given spectral efficiency. For BPSK, $\eta=1$, we see that theoretically SNR can be as low as 1, or 0 dB.

For the past fifty years, this channel capacity is never achieved by practical coding systems until Claude Berrou, Alain Glavieux and Punya Thitimajshima [BGT93] proposed Turbo Coding. Turbo coding is capable of attaining a channel capacity that is very closed to the limit as stated in equation 3.3.

3.3 Capacity of fading channel

Besides being corrupted by noise, the wireless media is also time varying. The Rayleigh distribution is commonly used to model the statistical time varying nature of the envelope of a flat fading channel, or the envelope of an individual multipath component [Pro89].

The probability distribution function (pdf) of Rayleigh distribution is given by

$$p(r) = \begin{cases} \dfrac{r}{\sigma^2} \exp(-\dfrac{r^2}{2\sigma^2}) & (0 \le \mathrm{r} \le \infty) \\ 0 & (\mathrm{r} \le 0) \end{cases} \tag{3.7}$$

where σ is the rms value of the received voltage signal before envelope detection and σ^2 is the time-average power of the received signal before envelope detection. The pdf of $y=r^2$, which is the power of the received signal, is given by

$$p(y) = \frac{1}{2\sigma^2} e^{-y/2\sigma^2} \tag{3.8}$$

which is chi-square-distributed with 2 degree of freedom.

The channel model for time varying channel corrupted with noise is given by

$$y=ax+n \tag{3.9}$$

where x is the transmitted signal, a is the fading amplitude while a Rayleigh distribution. n is an AWGN. The above channel can be instantaneously viewed as a Gaussian channel with signal voltage $r=ax$. From equation 3.3, we can find out the instantaneous channel capacity. However, owing to the time varying nature of the channel, the channel capacity of the Rayleigh channel must be calculated in an average sense.

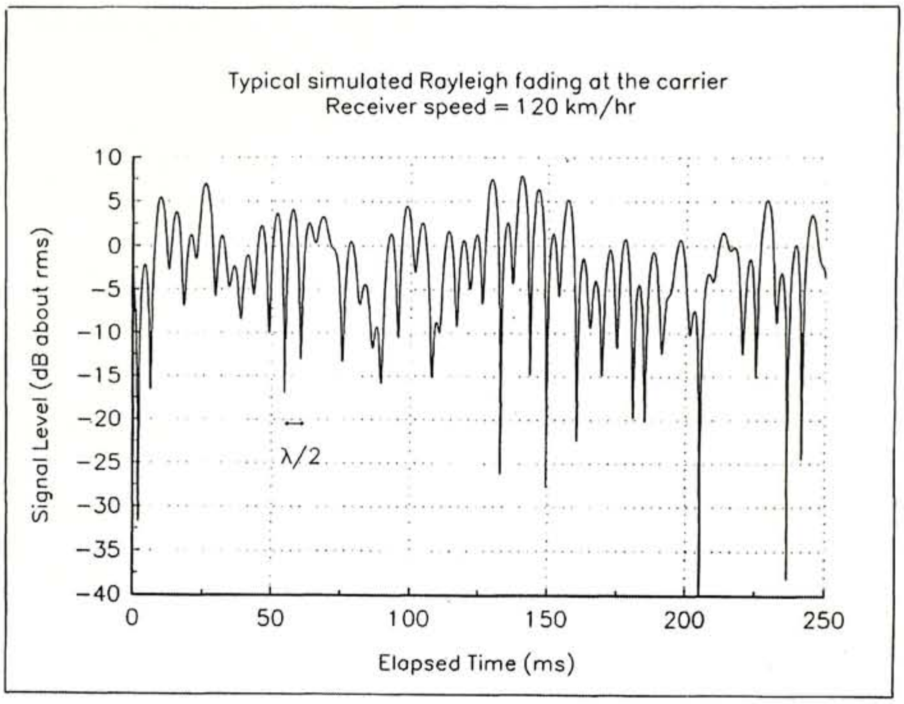

Fig 3.2 A typical Rayleigh fading envelope at 900 MHz [From [FRT93] copyright IEEE].

Lee [Lee90] has proposed a method to calculate Rayleigh channel capacity in the average sense. Suppose the carrier-to-noise ratio γ =C/N varies in time. From 3.8, we have the pfd of γ given by

$$p(\gamma) = \frac{1}{\Gamma} e^{-\gamma/\Gamma} \tag{3.10}$$

where Γ is the average power of γ, $\Gamma=<\gamma>=<C>/N$. The average channel capacity is

$$<C>=\int_0^\infty W\log(1+\gamma)\frac{1}{\Gamma}e^{-\gamma/\Gamma}d\gamma \tag{3.11}$$

$$=-W(\log e)(e^{-1/\Gamma})E_i(-\frac{1}{\Gamma}) \tag{3.12}$$

where E_i is the exponential-integral function and can be expressed in two different forms

$$E_i(-x)=E+\ln(x)+\sum_{k=1}^{\infty}\frac{(-x)^k}{k\cdot k!} \tag{3.13}$$

$$E_i(-x)=e^{-x}\cdot\sum_{k=1}^{n}(-1)^k\frac{(k-1)!}{x^k}+R_n \tag{3.14}$$

where $x>0$, E is the Euler constant ($E=0.5772157$), and R_n is the residual term. Put 3.13 into 3.12, we have

$$<C>=\frac{W}{\ln 2}\cdot e^{-1/\Gamma}\left[-E+\ln\Gamma+\frac{1}{\Gamma}-\frac{1}{(2\cdot 2!)\Gamma^2}+\frac{1}{(3\cdot 3!)\Gamma^3}-\frac{1}{(4\cdot 4!)\Gamma^4}+\ldots\right] \tag{3.15}$$

In case of $\Gamma>2$, equation 3.15 becomes

$$\frac{<C>}{W}=\log e\cdot e^{-1/\Gamma}(-E+\ln\Gamma+\frac{1}{\Gamma}) \tag{3.16}$$

From Fig 3.3, we see that it requires average carrier to noise ratio Γ at about 1.5 dB in order to have a capacity of 1 bit per second per Hertz.

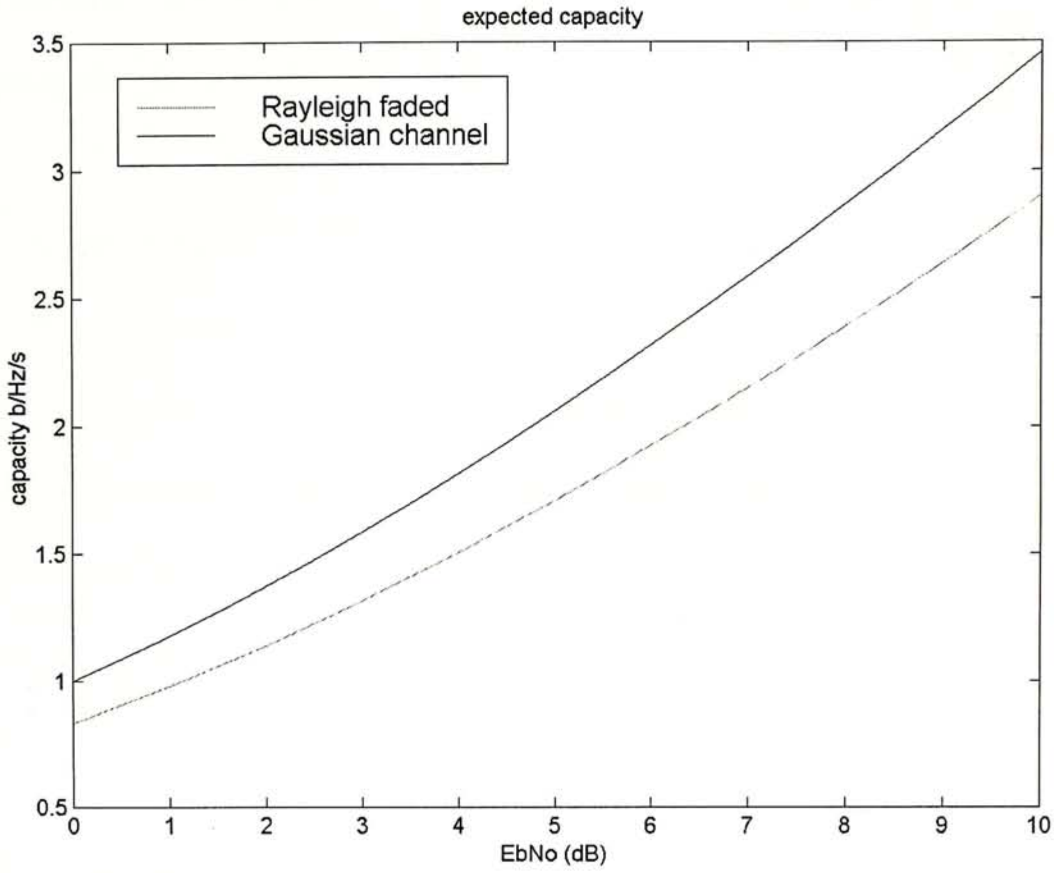

Fig 3.3 Channel capacity of Rayleigh channel and Gaussian channel

As shown in Fig. 3.3, Rayleigh fading channels have channel capacity closed to that of AWGN channels. However, capacity $<C>$ in equation 3.16 is obtained in an average sense. In case of a slow flat fading channel, a packet transmitted over the channel has an envelope almost constant for the whole packet. In such case, the condition of an individual signal undergoes Rayleigh fading independently is not justified. Thus, the average capacity <*C*> obtained in equation 3.16 can never achieved in slow flat Rayleigh fading channels.

Slow frequency hopping together with channel interleaving schemes can effectively combat slow flat Rayleigh fading. According to the channel interleaving scheme, received signals from different frequency bands are permuted into a block, where individual signals are faded according to the frequency bands they belong to. If the number of frequency bands is large enough, individual signals can be assumed to be independently faded. In theory, average channel capacity <*C*> in equation 3.16 is achieved. We can use that value to evaluate the performance of a practical system.

Turbo coding has been shown to have an excellent performance over the AWGN channel. However, turbo codes fail when applied over a flat Rayleigh channel. In chapter 4, we shall demonstrate that with the slow frequency hopping scheme, Turbo Codes with Iterative Channel Estimation technique are capable of combating the fading channel. We shall show that with large number of frequency channels, our scheme can have a performance fairly closed to the theoretical limit stated in equation 3.16.

Chapter 4

Iterative Channel Estimation for Turbo Code for Frequency Hopped Multiple Accessing

4.1 Introduction

PQMA adopts turbo coding for FEC. Many literatures have shown that turbo code [BGT93] has a good performance in Additive White Gaussian Noise (AWGN) channel. Some studies focus on the performance of turbo coding over Rayleigh faded channel

[HW96] [Jun96]. However, these studies have made the assumptions of a fully interleaved channel and perfect channel information in evaluating the performance.

We adopt turbo coding in PQMA to provide high bandwidth multimedia communication over wireless media. In PQMA, the message length of the turbo code is short, around 1000 bits. For the high bandwidth wireless channel, the channel is highly correlated over time. It is not possible to have the message bits "fully interleaved" within the short message length.

M.C Valenti, and B.D Woerner [VW98] have proposed some channel estimation by passing the received signal through an FIR filter to estimated the fading amplitude and the Gaussian property of the channel in order to improve the performance of turbo code applying over Rayleigh faded channel. In this chapter, we notice that turbo coding improves its accuracy after each decoding iteration. We use the decoded bit after each decoding iteration to estimate the channel state information. This is known as *Iterative Channel State Estimation.*

On the other hand, PQMA uses slow frequency hopping channel access to combat the highly correlated fading channel. We have simulated our system over different system configurations to see the performance of turbo code. In this chapter, we show that iterative channel state estimation can accurately estimate the channel state information and thus we can effectively adopt turbo coding to a highly correlated Rayleigh fading channel. We conclude that PQMA can provide different error correction capability that can meet the varieties of BER requirements for all the services.

4.2. Turbo code structures

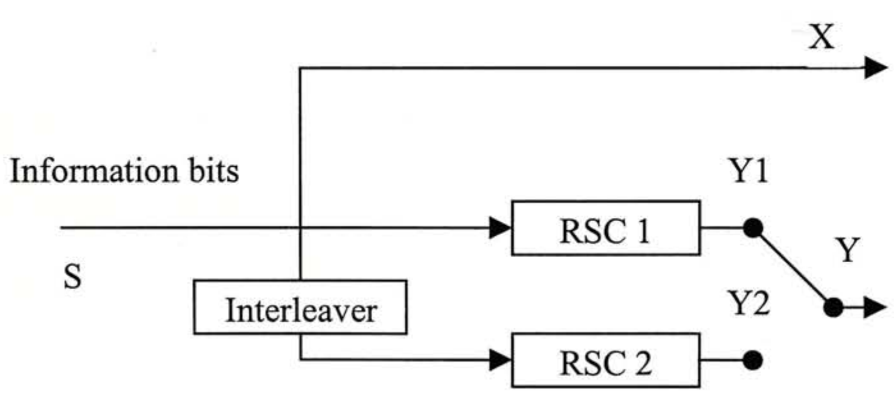

Fig. 4.1, Simplified Turbo code encoder.

Fig.4.1 shows a simplified turbo code encoder. Two identical RSC codes are concatenated in parallel. Both RSC encoders (RSC1 and RSC 2) use the same information bits but rearranged as different sequences due to the use of an interleaver. The code rates for RSC1 and RSC2 are the same. The parity streams *Y1* and *Y2* are alternatively punctured to yield *Y*. *X* is identical to the source information bits *S*. A generalized design of turbo code is to uses *n* encoding components parallel concatenated and *n-1* interleavers to provide permuted blocks for data bits of the encoders.

Turbo codes are decoded in an iterative manner. Decoding modules are put in a series for pipeline decoding. Turbo code decoding module at p^{th} iteration shown in Fig. 4.2 is made up of two elementary decoders DEC1 and DEC2 in a serial concatenation scheme. The first elementary decoder DEC1 is associated with encoder RSC1 and yield a soft (weighted) decision. The error bursts from DEC1 are scattered by the interleaver before passing to DEC2. Each elementary decoder use modified BAHL *et al.* algorithm [BGT93]. At the end of each decoding iteration, the Logarithm of Likelihood Ratio (LLR) z_p is passed for the $(p+1)^{th}$ decoding iteration. x_p and y_p are the input signal to decoder corresponding to information bits and parity bits respectively.

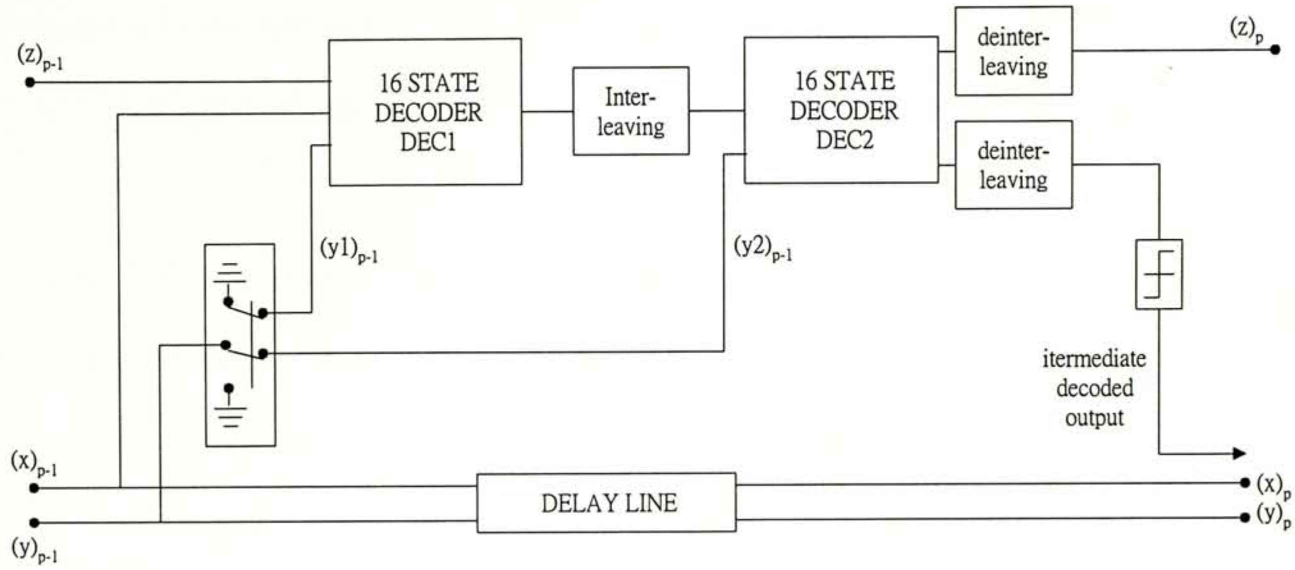

Fig. 4.2, Turbo code decoding module (pth iteration).

4.3 System Model

Wireless ATM is suitable for high bandwidth multimedia communication over the wireless environment. We design the system with an aim to adopt turbo coding in wireless ATM. For one 53-byte ATM cell, 5 byte is the header and 48 byte is the payload. We assume wireless ATM system treat the header and the payload in separated channels. We group 3 ATM payloads into one turbo code message block of 1152 bits. Referring to Fig. 4.3, the information bits, u_k are grouped into frames of 1152 bits. The frame is passed through a rate 1/2 constrain length three Turbo Encoder used in [BGT93] with code generator $(5/7)_8$. Random interleaving is employed in the encoder.

Since PQMA adopts SFH multiple access to combat fading, the 1152 bits message block is split into many sub-packets and each sub-packet is transmitted through different frequency channels. A channel interleaver interleaves the encoded bits, x_k, before they are packed into sub-packets. We can use a V by K block interleaver to serve as a channel interleaver, where V is the number of frequency channels and K is the sub-packet length. The interleaved bit stream, s_k, is packed into V sub-packets of K bits. Each sub-packet is hopped through a channel with frequency band i. For example, if there are 8 frequency channels in the system, then each sub-packet has 1152/8 = 144 bits. And the channel interleaver is a 144 x 8 block interleaver. Fig. 4.5 shows the performance of our system using block interleaver as channel interleaver. In our simulation, we have simulated 1, 4, 8, 16 and 32 frequency channels.

Another approach for channel interleaver is the random interleaver. In this case, the encoded bits, xk, are randomly permuted and then packed into sub-packets. Fig. 4.6 shows the performance of our system using random interleaver as channel interleaver.

The whole message block is 8192 bits and we have simulated 32 and 128 frequency channels.

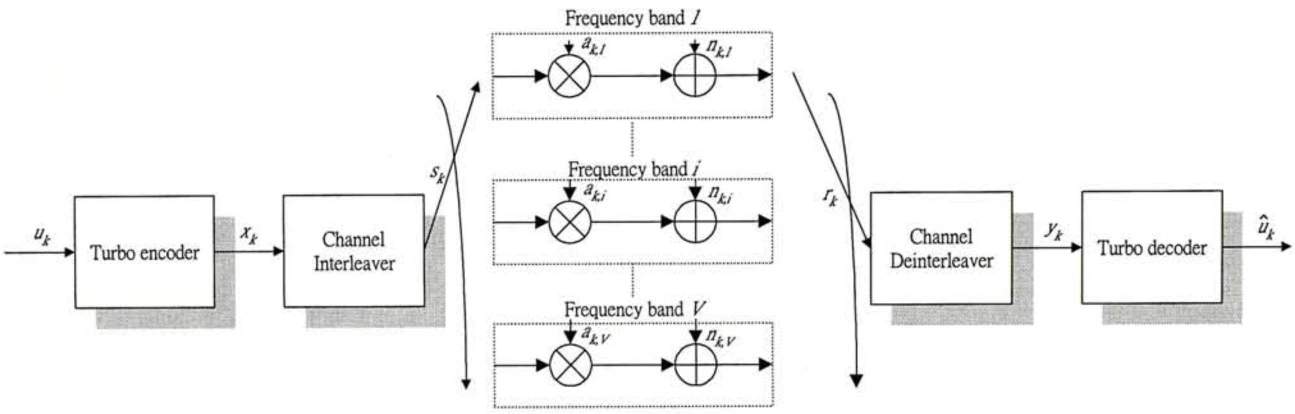

Fig. 4.3 System model of Turbo code over SFH multiple access

For broadband wireless communication, we model the channel with slow fading. We consider coherent BPSK signaling. The discrete representation of the channel i is given by $r_{k,i}=a_{k,i}s_{k,i}+n_{k,i}$ where k is the message index and i is the channel index. $s_{k,i}$ is a BPSK symbol. Since the channel is highly correlated and each sub-packet is very short, channel at frequency band i is Rayleigh faded with a fading amplitude $a_{k,i}$, which is assumed to be static over the whole sub-packet. An Addictive White Gaussian Noise (AWGN), $n_{k,i}$, with double-sided power spectral density (PSD) of N_0 is added to s_k. The received bit stream r_k is passed through a channel deinterleaver while becomes y_k. A turbo decoder using the

SOVA algorithm [PRV96] decodes the estimated information bits. The SOVA algorithm is modified to provide iterative estimation of the channel state estimation.

The symbol amplitude of $s_{k,i}$ is given by $\pm\sqrt{E_s}$ and the fading amplitude $a_{k,i}$ is represented in [Skl97] as:

$$p(a_{k,i}) = \begin{cases} 2a_{k,i}\exp\left[-(a_{k,i})^2\right] \text{for } a_{k,i} \geq 0 \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases}$$

The PSD N_0 of Addictive White Gaussian Noise is given by

$$N_0 / 2 = \sigma_n^2$$

In a Rayleigh faded channel, the turbo decoder must be modified in order to incorporate the channel state information. We assume that the fading amplitude is known to us from our iterative channel estimation. From [HW96] the transition probability is given by a conditional Gaussian distribution as

$$p(r_{k,i} \mid s_{k,i} = j, a_{k,i}) \sim N(a_{k,i}(2j-1)\sqrt{E_s}, N_o / 2)$$

for $j = 1$ *or* 0.

4.4 Iterative Channel Estimator

Iterative channel state estimation uses the decoded bits to estimate the channel states. At the end of each decoding iteration, SOVA algorithms can output the decoded bit steam of that iteration. Assuming the decoded bits are "correct", we use the difference between the decoded bits and the received bits to find out the error induced when the bits are transmitted through the channel. Using these error terms, we find out the channel states. When the number of iterations increases, the accuracy of the turbo decoded bits improves as well as the estimation of the error terms and of the channel states. On the other hand, since improved channel state estimation is passed to the turbo decoder for the next decoding iteration, the performance turbo decoder is also improved.

Since $r_{k,i}$, the received signal, is given by

$$r_{k,i} = a_{k,i} s_{k,i} + n_{k,i}$$

We can express

$$\begin{aligned} a_{k,i} &= \frac{r_{k,i} - n_{k,i}}{s_{k,i}} \\ &= \mathrm{sign}(s_{k,i})(r_{k,i} - n_{k,i}) \\ &= \mathrm{sign}(s_{k,i}) r_{k,i} - \mathrm{sign}(s_{k,i}) n_{k,i} \end{aligned}$$

where sign(.) equals to +1 or −1 accordingly for positive or negative operand. Now, since $a_{k,i}$ is static over the whole sub-packet and assuming $s_{k,i}$ is uniformly distributed over +1 and −1. For large N, with $n_{k,i}$ being a zero mean Gaussian random variable, we extend our estimation of the channel state after the *(j-1)*th iteration:

$$
\begin{aligned}
\hat{a}_{k,i}^{j} &= \frac{1}{N}\sum_{k=1}^{N} sign(\hat{s}_{k,i}^{j-1}) r_{k,i} - \frac{1}{N}\sum_{k=1}^{N} sign(\hat{s}_{k,i}^{j-1}) n_{k,i} \\
&\approx \frac{1}{N}\sum_{k=1}^{N} sign(\hat{s}_{k,i}^{j-1}) r_{k,i}
\end{aligned}
$$

which is the estimated fading amplitude for the j^{th} iteration. $\hat{s}_{k,i}^{j-1}$ is the intermediate decoded bit from the *(j-1)*th decoding iteration.

The variance of the channel is also an important factor for estimating the channel information. We propose to use iterative channel estimation technique in estimating the variance. The AWGN $n_{k,i}$ can be expressed as:

$$n_{k,i} = r_{k,i} - a_{k,i} s_{k,i}$$

and $n_{k,i}$ is roughly given by

$$n_{k,i} \approx r_{k,i} - a_{k,i} sign(\hat{s}_{k,i})$$

Again we use the intermediate decoded bits after the *(j-1)*th iteration and get,

$$n_{k,i}^{j} = r_{k,i} - \hat{a}_{k,i}^{j} sign(\hat{s}_{k,i}^{j-1})$$

where $\hat{a}_{k,i}^{j}$ is the estimated fading amplitude for the j^{th} iteration.

The statistical estimation of the noise variance for the j^{th} iteration is given by:

$$\begin{aligned}\left(\hat{\sigma}_n^2\right)_{k,i}^{j} &= \frac{1}{N-1}\sum_{k=1}^{N}\left(n_{k,i}^{j} - \mu_{k,i}^{j}\right)^2 \\ &= \frac{1}{N-1}\sum_{k=1}^{N}\left((r_{k,i} - \hat{a}_{k,i}^{j} sign(\hat{s}_{k,i}^{j-1})) - \mu_{k,i}^{j}\right)^2\end{aligned}$$

where

$$\mu_{k,i}^{j} = \frac{1}{N}\sum_{k=1}^{K}\left(r_{k,i} - \hat{a}_{k,i}^{j} sign(\hat{s}_{k,i}^{j-1})\right)$$

is the statistical mean of the noise.

In the next session, we shall show how these estimated channel state information, namely the fading amplitude a_k and the variance σ_k, are used in the turbo decoding algorithm.

4.5 Turbo decoding with iterative channel estimation

We use iterative channel estimation to provide the decoder with estimated channel information. Fig. 4.4 shows a modified turbo decoding module from [BGT93]. Turbo coding algorithms in DEC1 and DEC2 use channel reliability value to indicate the reliability of a particular received signal. The channel reliability of a received signal is given by $2a/\sigma^2$ for a AWGN channel with variance σ and fading amplitude a. The logarithm of likelihood ratio (LLR), L_all, associated with each decoded bit is given by

$$L\_all = \frac{2a}{\sigma^2} r_k + W_k$$

where W_k is a function of the redundant information introduced by the encoder.

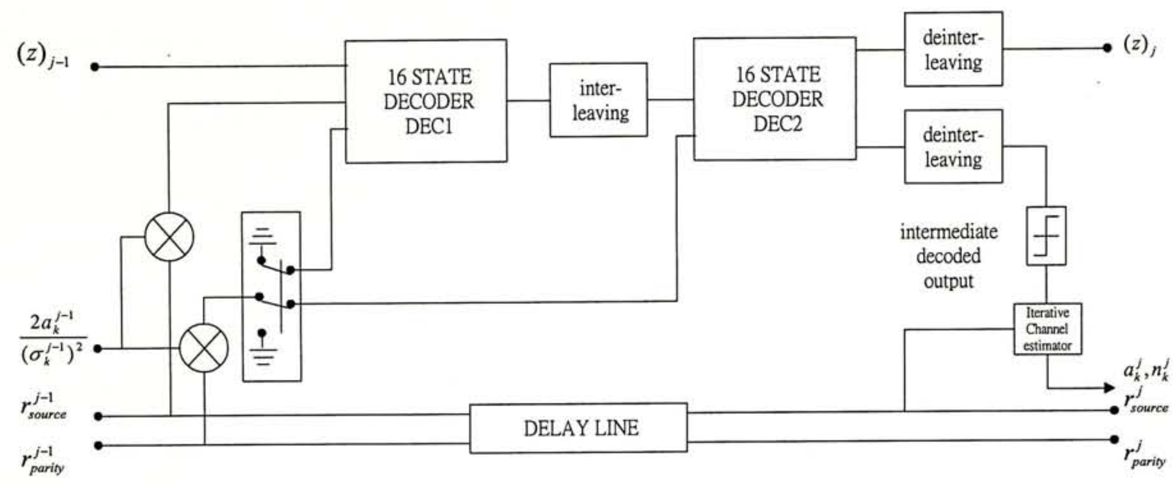

Fig. 4.4: Modified turbo decoding module (*j^{th}* iteration).

With iterative channel estimation, DEC1 and DEC2 obtained the reliability value from the Iterative Channel Estimator. After the *$(j-1)^{th}$* iteration, the reliability index, *L_c*, of a particular frequency channel is given by

$$L\_c = \frac{2a_k^{j-1}}{(\sigma_k^{j-1})^2}$$

The LLR of each decoded bit in j^{th} iteration is given by

$$L\_all = \frac{2a_k^{j-1}}{(\sigma_k^{j-1})^2} r_k + W_k$$

Referring to Fig. 4.4, in j^{th} iteration, the *$(j-1)^{th}$* module passes the received bit to the j^{th} module. Received bits are multiplied by the reliability index L_c before passing to the 16 state decoders. $(z)_{j-1}$ is the extrinsic value passed from (j-1)th module. After the decoding iteration, the intermediate decoded output is passed to the Iterative Channel Estimator to estimate the Channel State.

4.6 Simulation Results

The performance of turbo code over slow Rayleigh fading channels with iterative channel estimation was simulated. We have done simulations for 1, 4, 8, 16 and 32 frequency channels.

We use SOVA decoding algorithm with 8 decoding iterations in our simulations for the length of 1152 turbo codes. We can see that without frequency hopping as in the 1 channel case (marked with 1 in fig. 4.5.), turbo coding simply fails. From the turbo code performance in an AWGN channel, we can see that turbo codes have a very steep slope when the SNR increases from 1 dB to about 4 dB. If the SNR falls below 2 dB, turbo coding does not show a good coding gain. Under a slow flat fading environment, there is a chance that the entire a received packet has undergone deep fading. In such a case, the instantaneous SNR ratio is below 2 dB and turbo coding fails.

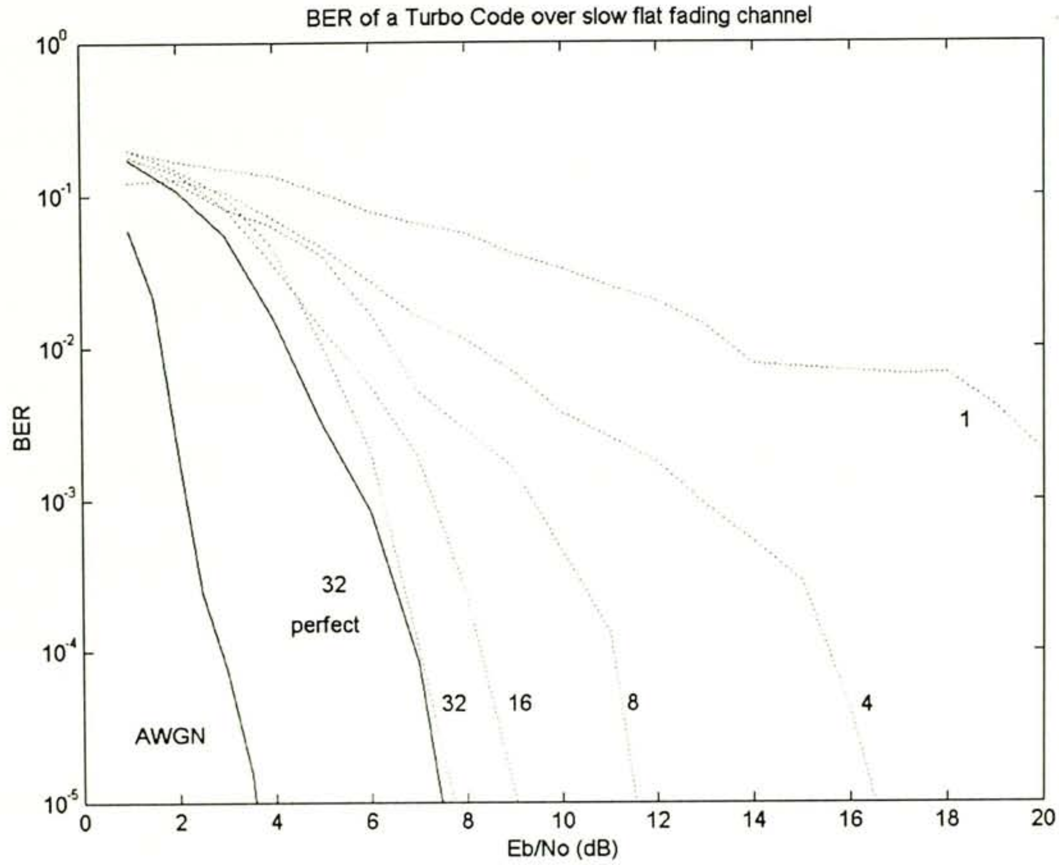

Fig. 4.5: Bit error performance of a turbo code of 1152 bit over slow Rayleigh fading channel.

When the whole packet undergoes deep fading, interleaving within the block itself does not improve the performance, as the fading amplitude is constant over the block. One way to deal with such a highly correlated channel is to increase the channel diversity by frequency hopping. Upon increasing the number of channels, turbo coding has improved performance. When the number of channel is increased to 16 with the length of each sub-packet being 144 bits, turbo code gives a BER of 10^{-5} at $E_b/N_o = 9$dB. When the number of channels is increased to 32 with the length of each sub-packet being 72 bits, it gives a BER of 10^{-5} at E_b/N_o less than 8 dB. We see that for message length K = 1152 bits of rate 1/2, the maximum number of channels is 32, and further increase in number of channels will have the length of sub-packet being very small. It is shown in Fig. 4.5 that the scheme of frequency hopping and channel interleaving can effectively improve the performance of turbo code over slow flat fading. It is shown that with 32 channel interleaving, turbo coding suffers only a 4 dB degradation over a slow flat Rayleigh channel, compared with the case of no fading.

We have simulated the case of turbo coding applied over a slow flat fading channel with 32 channel interleaving. We assume that the channel state information is perfectly known. The performance curve is shown in Fig. 4.5 marked with "32 perfect". We can see that Iterative Channel Estimation achieves a performance very close to perfect channel information. The performance degradation is less than 0.2 dB for medium SNR if iterative channel estimation is used instead of having perfect channel information.

E.K. Hall and S.G. Willson [HW96] have simulated the performance of turbo codes over highly correlated channel. For a channel with BT=0.001 using turbo code of rate = 1/3 with message length K=420, turbo code attains BER 10^{-5} at more than 20dB. With message length K = 5000, turbo code attains BER 10^{-5} at about 12 dB. Our result of 9dB with turbo code operating at code rate 1/2 and message length K = 1152 shows that frequency hopping with iterative channel estimation can effectively combat the slow fading channel.

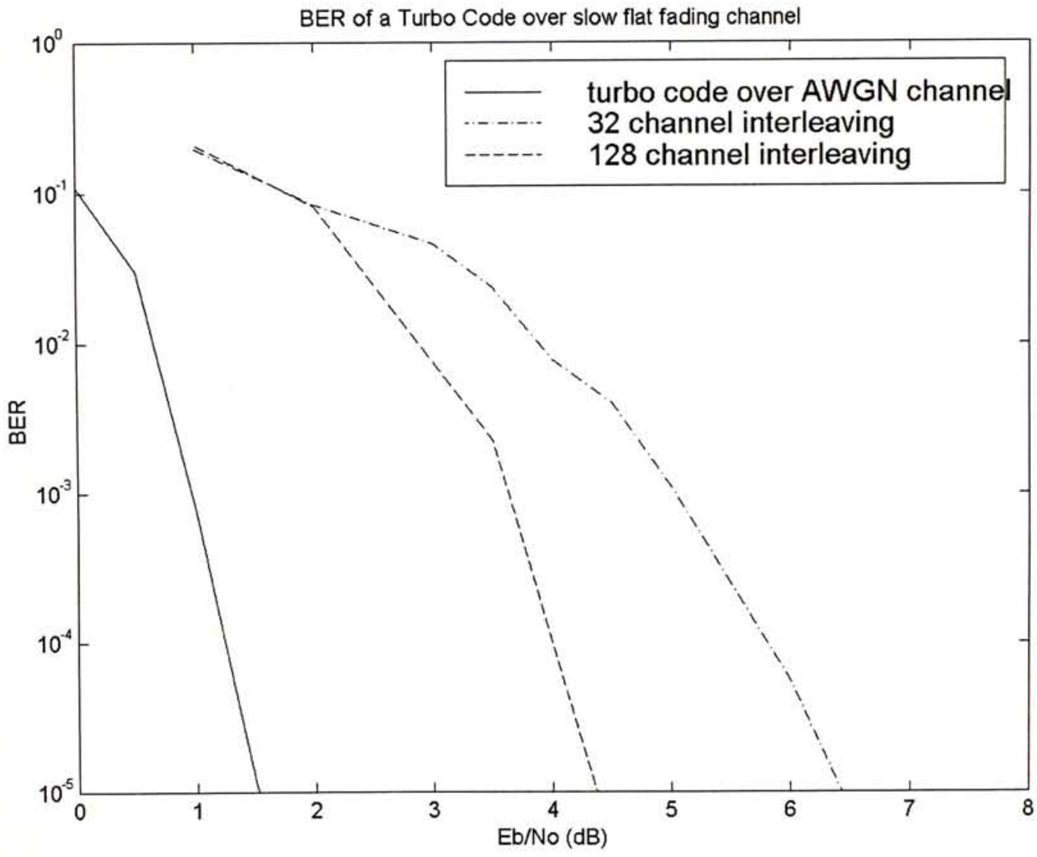

Fig. 4.6, Turbo code of 8192 bits applied over slow flat Rayleigh fading channel.

It is well known that turbo coding improves its performance when the interleaver size increases. We have simulated turbo codes for the length of 8192 bits using LOG MAP decoding algorithm and turbo coding is capable of achieving a BER of 10^{-5} at about 1.5 dB SNR in 8 iterations. The performance of turbo code over a slow flat Rayleight fading channel is shown in Fig. 4.6. We can see that for 32 channel interleaving, there is a 5 dB degradation due to slow flat Rayleigh fading. This is similar to the performance curves shown in Fig. 4.5 for turbo code of 1152 bits. However, when the number of channels is increased to 128, we can see that the degradation is decreased to about 3 dB. Turbo code can achieve a BER of 10^{-5} at about 4.5 dB SNR. In chapter 3, we have shown that theoretically to achieve channel capacity of 1, Raleigh fading channels bring a 1.5 dB SNR degradation when compared with AWGN channels. Therefore, degradation of 3 dB SNR in our proposed scheme achieves a performance fairly close to the theoretical limits.

4.7 Conclusion

In this chapter, we have shown by simulation that frequency hopping together with channel interleaving is capable of combating slow flat Rayleigh fading. Our proposed scheme has shown a performance fairly closed to the theoretical limits on the performance of a slow flat Rayleigh fading channel. In addition, our proposed iterative channel estimation can accurately estimate the channel state information. The degradation using our estimation is only 0.2 dB SNR when compared with using perfect channel information. Therefore, PQMA has shown a strong error correction capability and is thus suitable for handling the variety services in wireless multimedia communications.

Chapter 5

Dummy Bits Inserted Turbo code

5.1 Introduction

Error correction schemes are commonly used in many systems in order to provide reliable communications. Parallel concatenated convolutional coding (PCCC) or turbo code achieves a very impressive coding gain when applied to an AWGN channel [BGT93].

We adopt turbo coding for FEC in PQMA. Turbo coding uses an interleaver to reduce the multiplicity of the pairing of low-weigh codeword. Many researches have focused on the design of interleavers in order to improve the performance of turbo coding [PHBB95] [AHK98].

It has been shown that large interleaver size is an important issue of reducing the multiplicity of low-weight codewords. Performance of turbo coding decreases with a decrease in interleaver size. PQMA is designed to be a wireless ATM system. An ATM cell payroll has only 384 bits. The interleaver size for turbo coding in PQMA is therefore very short if we limit the interleaver size to 384 bits. In this chapter, we propose a new coding scheme that aims to improve the performance of turbo coding by the use of dummy bits to reduce the multiplicity of low-weight codewords. Dummy bits are inserted into the block of source data symbols. Since dummy bit insertion is independent of the length of the source block, our proposed technique can effectively reduce the multiplicity of low-weight codewords even in a message block of short frame size. In this chapter, we have simulated the proposed system to see if it improves the performance of turbo coding.

5.2 Weight Distribution of turbo codes

The component codes used in turbo coding are Recursive Systematic Codes (RSC). In order to estimate the performance of turbo coding, we must have information of the minimum distance, weight distribution or the actual code geometry of the codes [DF95].

For a RSC code using the $(5/7)_8$ code generator, if an all zero input u=(00…000…00) is encoded, the parity bits will be all zero too, i.e., parity bits are y=(00…000…00). Since RSC codes are linear, we can use the error pattern in decoding an all zero codeword to analyze the performance of the codes.

We now focus on the investigation of low-weight inputs. Since RSC use an Infinite Impulse Responds (IIR) code generator, the encoder will not return to the all zero state if a weight-1 message is input to the encoder. Therefore, u=(00…001000…00) is not a low-weight codeword as the recursive encoder will generator a codeword with an infinite number of ones in the parity bits.

The weight-2 inputs are particularly important in the study of performance of turbo coding. For u=(00…001001000…00), the recursive encoder will return to the all zero state by the input of the second bit of 1. The parity bits are y=(00…001111000…00). The resulting Humming distance of this codeword is 6. It turns out that weight-2 inputs with the following pattern will drive the encoder to the all zero state:

$$u=(00\ldots001\underbrace{00\ldots000}_{2+3t \text{ zeros}}100\ldots0$$

That is, for an integer *t* greater than zero, a weight-2 input with $2+3t$ zeros separating the two 1's would drive the encoder to the all zero state. If *t* is not an integer but equals to multiples of 1/3, the encoder will not return to the all zero state.

For a rate 1/3 turbo code having two encoding streams, an interleaver rearranges the sequence of the input bits for the encoder 2. If a weight-2 input with its 1's separated by $2+3t_1$ zeros in encoder one is interleaved to form a weight-2 input with its 1's separated by $2+3t_2$ zeros for encoder 2, the total Hamming weight of the turbo code will be $2+2*4+3(t_1+t_2)$. For the worst case, $t_1=t_2=1$, the Hamming weight equals 10. With the use of an interleaver, the probability of having such worse cases is minimized.

To explain the function of an interleaver, we now consider a rate 1/3 turbo code with interleaver size K=100. Let p be the probability of having a weight-2 input with its 1's separated by two zeros, i.e., u=(00…001001000…00), is interleaved to a weight-2 input with its 1's also separated by two zeros for the second encoder. Assuming the interleaver randomly permutes the input bits, there are totally ${}_{100}C_2 = 4950$ possible permutations for the two 1's. Among all these permutations, only 97 permutations will have the two 1's separated by two zeros. Therefore, the probability p is given by 97/4950 = 0.02.

Since the probability of the occurrence of pairing of low-weighted codewords is greatly reduced, the use of an interleaver can effectively improve the performance of turbo coding. In addition, this is the reason why turbo coding improves its performance with an increase in interleaver size.

With the use of dummy bits inserting into the message block, we aim to reduce the multiplicity of the occurrence of low-weight codewords in each component encoder. In session 5.5, we will discuss how the goal can be achieved.

5.3 Encoding with dummy bit insertion

Both the transmitter and receiver have complete knowledge of the dummy bits. We insert these dummy bits into the source data block. Let S the original data block of length N and let D be the dummy bits of length M. Now, $S=(s_1, s_2, \ldots, s_N)$ and $D=(d_1, d_2, \ldots, d_M)$. Define R as the *insertion ratio* where $R=N/M$. For every bit in the source data block, there are R dummy bits inserted into the data block. Let X_I denote the data block with dummy bits inserted into the source data. We pass X_I to the turbo encoder as described in section 4.2.

5.3.1 Dummy bit insertion methodology

Dummy bits act as anchor points in the codeword. We define the number of information bits between two consecutive dummy bits as the span of the dummy bits. We evenly distribute the dummy bits over the codeword so as to maximize the dummy bit spanning of the information stream. We call this a block insertion.

Case I.) $R \geq 1$

For $R \geq 1$, then $M \geq N$, we divide the dummy bits into N blocks of length R. Then,

$X_I=(s_1, d_1, \ldots, d_R, \ldots, s_k, d_{(k-1)R+1}, \ldots, d_{kR}, \ldots, s_N, d_{(N-1)R+1}, \ldots, d_{NR})$

Case II.) $R<1$

If $R < 1$, then $M<N$, we divide the source data block into M portions. One dummy bit is inserted for each portion. We have

$X_I=(s_1,.., s_{\lfloor 1/R \rfloor}, d_1, ..., s_{\lfloor k/R \rfloor}, d_k, ..., s_N)$

where $\lfloor \cdot \rfloor$ denotes the rounding down operator.

5.3.2 Hybrid Periodic Random Interleaver

As mentioned before, dummy bits are evenly distributed in the inserted information stream X_i. X_i is permuted and encoded by RSC 2. If random interleaving is used for the permutation, the dummy bits will also be randomly permuted in the inserted information stream for RSC 2. The dummy bits in the permuted inserted information stream are not distributed evenly. In this case, the dummy bit span is not maximized. In order to keep the dummy bits evenly distributed on the inputs for the second encoder, a hybrid periodic random interleaver is needed.

The hybrid periodic random interleaver consists of random interleaving and a periodic block insertion. Information stream and dummy bits are independently interleaved using random interleavers. The two permuted sequences are merged together using periodic block insertion method mentioned in session 5.3.1.

5.3.3 Dummy bit removal before transmission

Since the receiver has complete knowledge of the dummy bits (for example, the dummy bits are all zero), there is no need to send the dummy bits over the channel. Therefore, as shown in Fig. 5.1, the Dummy Bit Remover (DBR) punctures the dummy bits before transmission.

The Dummy bits together with information bits are encoded by the encoder, therefore, dummy bits introduce extra parity bits. Since the dummy bits are known to the decoder, therefore, we may not need the extra parity bits to protect the dummy bits. As shown in Fig. 5.1,the Extra Parity Bit Remover (EPBR) removes these extra parity bits. It is clear to see that by puncturing all the dummy bits and extra parity bits introduced by them, the overall code rate of the code is not changed.

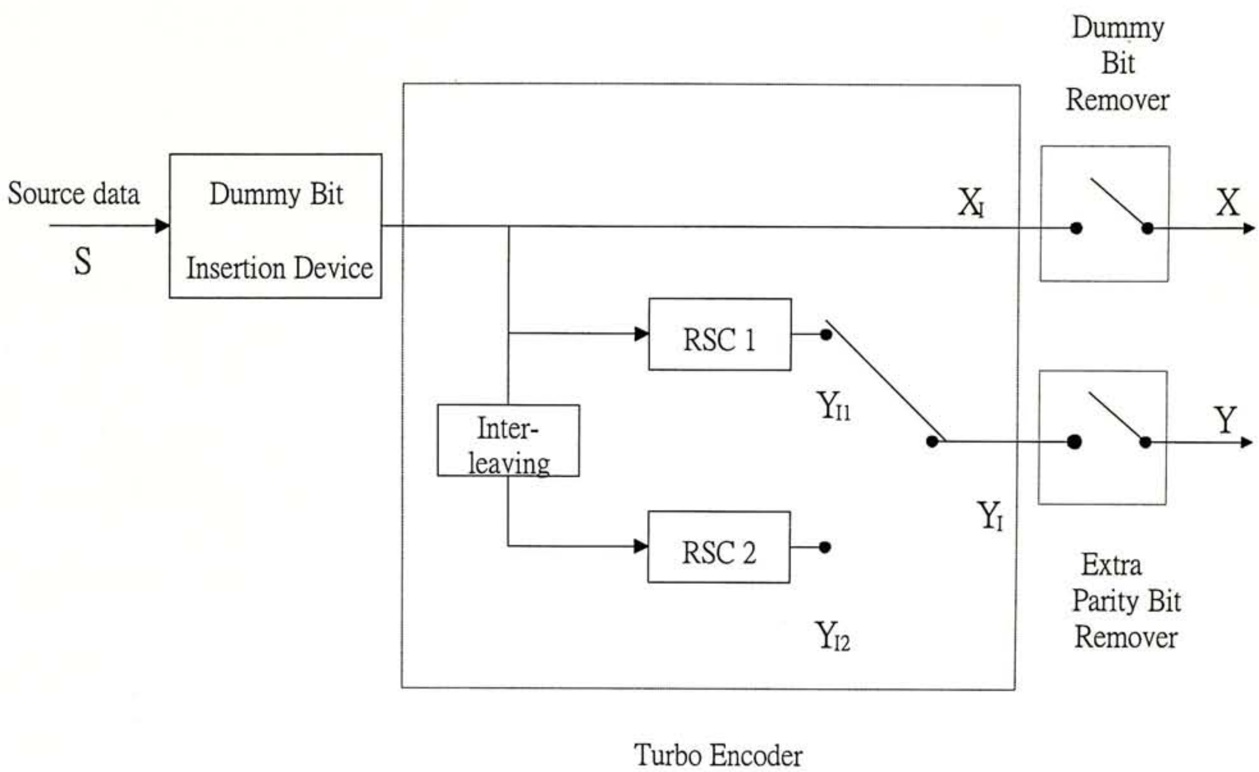

Fig. 5.1, encoding process with dummy bit insertion.

5.4 Decoding with dummy signal enhancement

In the decoding part of dummy bit inserted turbo coding, we do not modify the decoding algorithms used in classical turbo decoding. What we have to do is to reconstruct the dummy signals in the received signal stream and fill the position of the punctured extra parity bits with zeros.

The Dummy Signal Insertion Device (DSID) in fig. 5.2 inserts dummy signals to x according to the positions of the dummy bits in the inserted information stream. For the sake of simplicity, assume that the energy per bit used in our system is 1. The amplitude of the dummy signal is set to +1 or −1 for the dummy bits of 1 or 0 accordingly.

Turbo coding algorithms use channel reliability values to indicate the reliability of a particular received signal. The channel reliability of a received signal is given by $2a/\sigma^2$

for a AWGN channel with variance σ and fading amplitude a. The LLR associated with each decoded bit is given by

$$L\_all = \frac{2a}{\sigma^2} r_k + W_k$$

where W_k is a function of the redundant information introduced by the encoder.

Since dummy bits are not corrupted by noise, they are very "reliable". Therefore, the channel reliability value of a dummy bit should be very large so that the decoder will never output a wrong value of the dummy bit. We note that the channel reliability value is proportional to the fading amplitude a. We therefore scale up the signal amplitude of the dummy signals to "tell" the decoders that dummy signals are reliable.

Dummy Signal Amplifier (DSA) depicted in fig. 5.2 scales up the signal amplitude of the dummy signals. For a *dummy signal enhance ratio A*, where $A >> 1$, the LLR associated with a dummy bit is given by

$$L\_all = \frac{2A}{\sigma^2} r_k + W_k$$

where W_k is a function of the redundant information introduced by the encoder.

Since we have punctured the extra parity bits introduced by dummy bits, we have to fill the corresponding position with 0's in the parity streams of the received signals. In fig. 5.2, Extra Parity Signal Insertion (EPSI) unit performs this task.

The dummy signal processing modules, namely, Dummy Signal Insertion Device, Dummy Signal Amplifier and Extra Parity Signal Insertion are only required at the beginning of the decoding process. That is, they are not required in turbo decoding modules other than the first iteration.

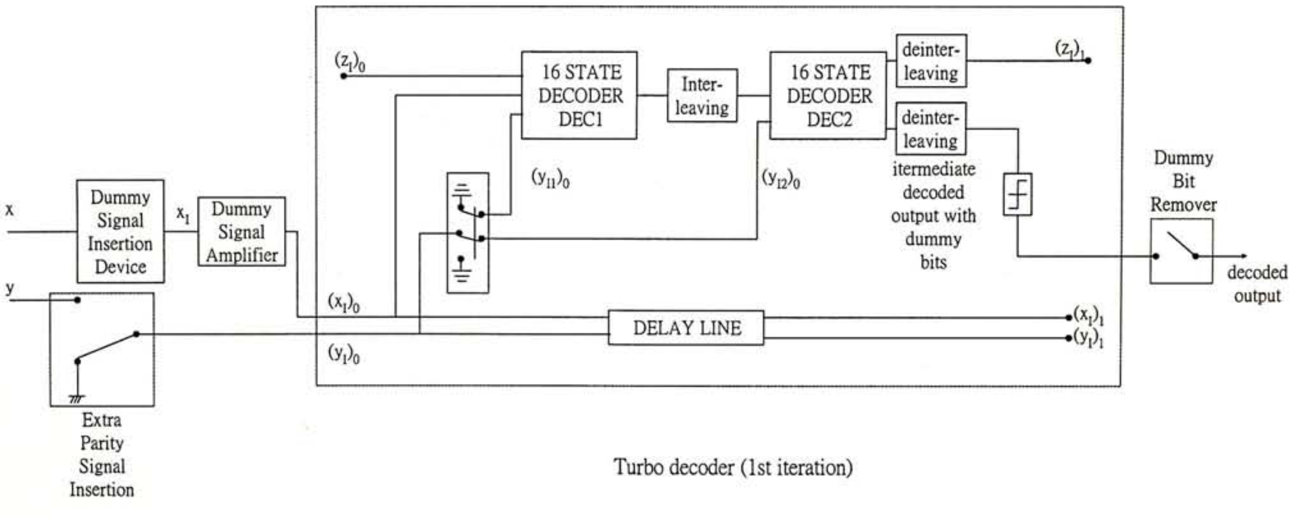

Fig. 5.2, decoding process for dummy signal enhancement

5.5 Weight distribution of dummy bit inserted turbo coding

Now we study the case if we use dummy bit insertion with an insertion ratio 1/3 and assume that we set all dummy bits to 0. For an all zero message block, after dummy bit insertion, we have (000**0**000**0**000…), where the "**0**" in bold face denotes a dummy bit. We can see that low-weight codeword can only occur in some positions.

For example, the low-weight codeword (100**1**000**0**000…) is not a "valid" codeword because decoder will not output the forth bit "**1**" as dummy bit forces it to be "**0**". In this case, we reduce the multiplicity of low-weight codewords.

However, low-weight codeword (010**0**100**0**000…) with parity bits (011**1**100**0**000…) is valid. Since we puncture the parity bits generated by dummy bits, which is the fourth bit "**1**" and the eighth bit "**0**" in this example, the total Hamming distance of this codeword is reduced to 5, instead of 6.

In dummy bit inserted turbo coding with insertion ratio 1/3, if weight-2 input with its 1's separated by two zeros is randomly interleaved, there are only 64 possible permutation that will result in a weight-2 input with its 1's separated by two zeros. The probability of have pairing of two low-weighted codewords is then 64/4950=0.015.

Avoiding the multiplicity of low-weight codeword will improve the performance of turbo codes. However, reduction of Humming distance will lead to degradation in performance. Intensive simulations have been done to see if variation of dummy bit insertion pattern can improve the performance of turbo coding.

5.6 Simulation results

We have simulated our proposed system. In order to investigate the performance of turbo coding over short frame system, we compare turbo codes of length 100 bits. Constrain length three Turbo Encoder of rate 1/3 used in [BGT93] with code generator $(5/7)_8$ is used as a control simulation. Random interleaving is employed in the encoder.

For turbo codes with dummy bits inserted to information stream, the dummy bit insertion ratio is 1, which means every information bit is associated with one dummy bit. Dummy signal enhance ratio *A=128*. The code rate is 1/3 and code generator used is $(5/7)_8$. Fig. 5.3 shows the BER performance of the codes. Inserting dummy bits into the information block cannot improve the performance of turbo codes. On the other hand, there is a little degradation recorded.

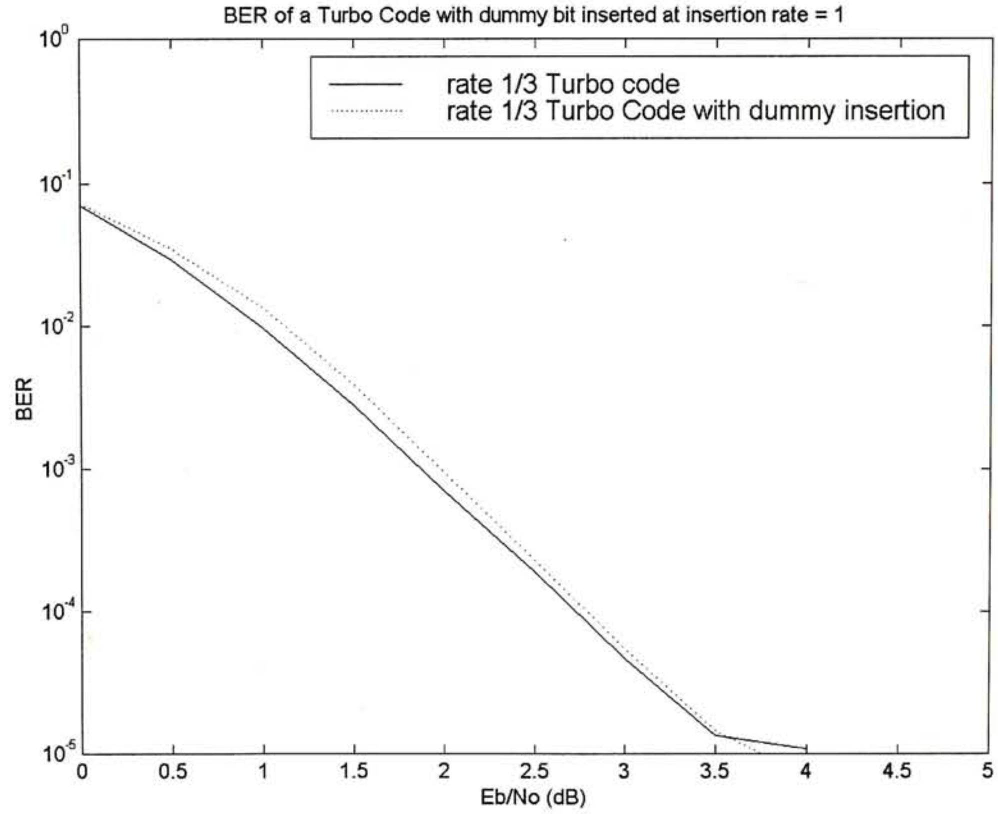

Fig. 5.3, BER of turbo code with dummy bits insertion ratio 1.

5.7 Summary

In this chapter, we have proposed to insert dummy bits into the information block to reduce the multiplicity of low-weight codeword of turbo codes. However, since puncturing the parity bits introduced by dummy bits reduces the free distance of the constituent RSC code, the overall performance of the proposed coding scheme has shown a small degradation as compared to the classical turbo code.

At this phase of study, simple dummy bit insertion patterns fail to improve the performance of turbo coding. However, we have demonstrated that using dummy bits is a feasible method of attacking the multiplicity of low-weight codewords. Finding a non-trivial dummy bit insertion patterns that can improve the performance of turbo code can be a topic for further research.

In this thesis, we have proposed PQMA for wireless ATM, accommodating multimedia services and support distributed multiple access with flexible bandwidth assignment. We have demonstrated that adopting turbo coding with slow frequency hopping multiple channel access can effectively combat slow flat Rayleigh fading channels. Using Iterative

Channel State Estimation technique to provide channel state information for turbo coding, we have shown a performance very close to using perfect channel information. We conclude that PQMA adopting RCPTC can effectively support ATM services over wireless media.

References

[AEK96] E. Ayanoglu, K. Eng, M. Karol, "Wireless ATM: Limits, Challenges, and Proposals," IEEE Personal Communications, Vol. 3, no 4. Aug. 1996.

[AHK97] K. S. Andrews, C. Heegard, and D. Kozen. "A theory of interleavers. Technical Report TR97-1634, Department of Computer Science, June 1997.

[AN94] A.S. Acampora, M. Nagshineh, "An Architecture and Methodology for Mobile-Executed Handoff in Cellular ATM Networks," IEEE JSAC, vol. 12 Oct. 1994.

[BGT93] C. BERROU, A. GLAVIEUX and P. Thitimajshima, "Near Shannon limit error-correcting ocding and decoding: Turbo-Codes", In ICC, pages 1064-1070, 1993.

[DF95] D. Divsalar and F. Pollara, "Turbo Codes for PCS Applications," IEEE 1995.

[FRT93] Fung, V., Rappaport, T. S., and Thoma, B., "Bit Error Simulation for $\pi/4$ DQPSK Mobile Radio Communication using Two-ray and Measurement-based Impulse Respone Modles," IEEE Journal on Selected Areas in Communication, Vol. 11 No. 3, pp. 393-405, April 1993.

[FYP96] Cheng Siu Lung, "Predictive Queuing Multiple Access (PQMA) – a media access protocol for wireless ATM," Final Year Project Report, Information Engineering Department, The Chinese University of Hong Kong, June 1996.

[Hag96] J. Hagenauer, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Info Theory,* vol. 42, no. 2, Mar. 1996, pp. 429-45

[HW96] E.K. Hall and S.G. Willson, "Design and performance analysis of turbo code on Rayleigh fading channels," in *Proc., CISS,* (Princeton NJ, 1996)

[JN96] M. Jordan and R. Nichols, "The effects of channel characteristics on turbo code performance," in *Proc., MILCOM*, (McLean VA, 1996), pp. 17-21.

[JPDB97] Jung, P., Plechinger, J., Doetsch, M., Berens, F.M. "A Pragmatic Approach to Rate Compatible Punctured Turbo-Codes for mobile radio applications." 6th International Conference on Advances in Communiations and Control: Telecommunications/Signal Processing, Grecotel Imperial, Corfu Greece, 23-27 June 1997.

[Jun96] P. Jung, "Comparison of turbo-code decoders applied to short frame transmission systems," *IEEE J. Select. Areas Commun.,* vol. 14, pp. 530-7, Apr. 1996

[KM97] Osama Kubbar and Hussein T. Mouftah. "Multiple Access Control Protocols for Wireless ATM: Problems Definition and Design Objectives," IEEE Communications Magazine, November 1997.

[KS94] G. Kaplan and S. Shamai, "Achievable performance over the correlated Rician channel," *IEEE Trans. Commun.*, vol 42, pp. 2967-78, Nov. 1994

[Lee90] William C. Y. Lee, "Estimate of Channel Capacity in Rayleigh Fading Environment," IEEE Transactions on Vehicular Technology, Vol. 39, No. 3, August 1990.

[PHBB95] Podemski, Holubowicz, Berrou, Battail. "Hamming Distance Spectra of Turbo-Codes." Annales Telecommunication, 50(9):790-797, 1995.

[Pro89] John G. Proakis. "Digital Communications" Second Edition, McGraw-Hill International Editions, 1989.

[PRV96] L. Papke, P. Robertson, and E. Villebrun, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc., IEEE Int. Conf.* On Commun., (Dallas TX, 1996), pp. 102-6.

[Rob94] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in *Proc. IEEE CLOBECOM Conf.,* 1994, pp. 1298-1303.

[RW92] D. Raychaudhuri, N.D. Wilson, "ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks," IEEE JSAC, vol. 12, Oct. 1992.

[Sha48] C. E. Shannon. "A mathematical theory of communication," Bell System Technical Journal, 27:379-423, 623-656, October 1948.

[Skl97] B. Sklar, "Rayleigh fading channels in mobile digital communication systems part 1: Characterization," *IEEE Commun. Mag.,* vol. 35, no. 7, pp. 90-100, July 1997

[Toh97] Toh, C.-K. "Wireless ATM and AD-HOC networks : protocols and architectures," Kluwer Academic, c1997.

[VW98] M.C Valenti, and B.D Woerner, "Performance of Turbo Codes in Interleaved Flat Fading Channels with Estimated Channel State Information", IEEE 1998.

[WROL99] Francis Swarts, Pieter van Rooyan, Ian Oppermann, Michiel P. Lotter, "CDMA Technologies For Third Generation Mobile Systems", Boston : Kluwer Academic Publishers, 1999.