

## INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
800-521-0600

UMI<sup>®</sup>





**RELIABILITY ANALYSIS OF ATM NETWORK  
WITH ERROR CONTROL**

BY  
**MAHFOOZ SALEH BIN MAHFOOZ**

A Thesis Presented to the  
DEANSHIP OF GRADUATE STUDIES  
**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**  
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the  
Requirements for the Degree of

**MASTER OF SCIENCE**  
In  
**ELECTRICAL ENGINEERING**

**MAY 2001**

UMI Number: 1404201

UMI<sup>®</sup>

---

UMI Microform 1404201

Copyright 2001 by Bell & Howell Information and Learning Company.

All rights reserved. This microform edition is protected against  
unauthorized copying under Title 17, United States Code.

---

Bell & Howell Information and Learning Company

300 North Zeeb Road

P.O. Box 1346

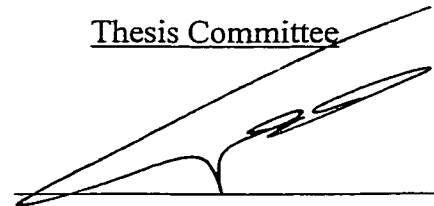
Ann Arbor, MI 48106-1346

**KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS  
DHAHRAN 31261, SAUDI ARABIA**

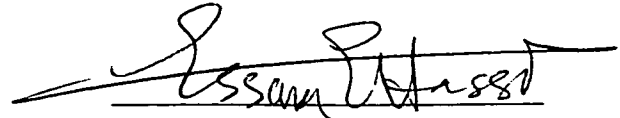
**DEANSHIP OF GRADUATE STUDIES**

This thesis, written by **Mahfooz Saleh Bin Mahfooz** under the direction of his Thesis Advisor and approved by his Thesis Committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**.

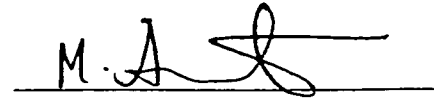
Thesis Committee



Dr. Maan A. Kousa (**Chairman**)



Dr. Essam Hassan (**Member**)



Dr. Mohammed A. Khan (**Member**)



**Department Chairman**



**Dean of Graduate Studies**



13/5/2001

Date

DEDICATED TO MY FATHER AND MOTHER

## **ACKNOWLEDGMENT**

Acknowledgment is due to King Fahd University of Petroleum & Minerals for supporting this work.

I would like to express my appreciation to Dr. Maan Kousa, my principal advisor, for his careful guidance and encouragement through this research.

I am also grateful to other committee members Dr. E. Hassan and Dr. M. Khan for their useful and valuable suggestions.

Thanks are also due to all friends who helped and/or encouraged me during the work in this thesis. Among all, many thanks to Osamah Faris, Mohammed Abu-Sada and Ahmed Nawajha.

Last but not least, thanks are due to all my family members for their patience, support and sacrifice.

# TABLE OF CONTENTS

	<b>Page</b>
Table of Contents .....	v
List of Figures .....	vi
List of Symbols .....	vii
Abstract (Arabic).....	ix
Abstract (English) .....	xi

## **CHAPTER 1**

### **Introduction and Literature Review**

1.1 ATM Networks .....	2
1.1.1 ATM Technology.....	3
1.1.2 ATM Characteristics .....	4
1.1.3 ATM Protocol Hierarchy .....	5
1.1.4 ATM Cell Structure .....	7
1.1.5 ATM Switching .....	10
1.1.5.1 Input Buffering.....	11
1.1.5.2 Output Buffering .....	13
1.1.5.3 Shared Buffering .....	13
1.1.6 Congestion in ATM .....	14
1.1.7 Wireless ATM.....	14
1.2 Channel Coding .....	16
1.2.1 Error correction and detection.....	17
1.2.2 Error Control Coding .....	18
1.2.2.1 CRC codes .....	21
1.2.2.2 Parity-Check Codes.....	23
1.3.2.2 Product Codes .....	23
1.2.3 Erasure Decoding.....	26
1.3 Literature Review.....	27
1.4 Proposed work .....	32



## **CHAPTER 2**

### **RELIABILITY STUDY: INFINITE BUFFER CASE**

2.1	Introduction.....	35
2.2	Random channel case.....	40
2.2.1	Pre-decoding probabilities .....	40
2.2.2	Post-decoding probabilities.....	41
2.2.3	Discussion and results.....	44
2.2	Correlated channel case .....	50
2.2.1	Simulation of correlated channel .....	53
2.2.2	Discussion and results.....	56
2.4	Simulation results .....	64
2.5	Summary .....	68

## **CHAPTER 3**

### **RELIABILITY STUDY: FINITE BUFFER CASE**

3.1	Introduction.....	69
3.2	Random traffic loading .....	71
3.2.1	Pre-decoding probabilities .....	71
3.2.2	Post decoding probabilities .....	75
3.2.3	Discussion and results.....	76
3.3	Bursty traffic loading .....	87
3.3.1	Input traffic model .....	87
3.3.2	Simulation.....	91
3.3.3	Discussion and results for random channel.....	94
3.3.4	Discussion and results for correlated channel.....	107
3.4	Summary .....	116

## **CHAPTER 4**

	<b>Conclusions and Suggestions for Further Work.....</b>	<b>117</b>
	<b>References.....</b>	<b>120</b>

## List of Figures

Figure	Pages
1.1 ATM Network Layers .....	6
1.2 ATM Cell format.....	7
1.3 Structure of the Header of ATM cells for both interfaces.....	9
1.4 ATM Switch schematic.....	12
1.5 Buffering strategies of ATM switches .....	12
1.6 General shape of product codes .....	25
1.7 Example of a single parity product code.....	25
2.1 Configuration of the FEC model.....	36
2.2 Format of the coding matrix.....	37
2.3 The Data and Parity cell formats.....	37
2.4 Lost cell regeneration .....	39
2.5 Effect of matrix size on cell loss rate .....	45
2.6 Cell error rate for 16×16 matrix size.....	47
2.7 Comparison of row-column-wise decoding and BIII in [34] .....	49
2.8 Two state Markov process correlated channel model .....	51
2.9 Study of channel correlation coefficient effect .....	52
2.10 Flow chart of correlated channel .....	55
2.11 Comparison of random channel and correlated channel for pre-decoding cell loss rate .....	57
2.12 Comparison of random channel and correlated channel for pre-decoding cell error rate .....	58
2.13 Simulation of correlated channel with correlation coefficient 0.0 (pre- decoding cell loss rate).....	59
2.14 Simulation of correlated channel with correlation coefficient 0.0 (pre- decoding cell loss rate).....	60
2.15 Cell loss rate for correlated and random channels .....	62
2.16 Cell error rate for correlated and random channels.....	63

2.17	Simulation results of cell loss rate for random channel .....	64
2.18	Simulation results of cell error rate for random channel.....	65
2.19	Simulation results of cell loss rate for correlated channel .....	66
2.20	Simulation results of cell error rate for correlated channel.....	67
3.1	Output Buffered ATM switch .....	70
3.2	Cell loss rate for random channel with infinite buffer .....	78
3.3	Cell loss error for random channel with infinite buffer .....	79
3.4	Effect of buffer size on cell loss rate.....	80
3.5	Effect of traffic intensity on cell loss rate .....	81
3.6	Cell loss rate versus buffer size.....	83
3.7	Cell loss rate for correlated channel with finite buffer size .....	84
3.8	Simulation of cell loss rate for random channel with finite buffer .....	85
3.9	Simulation of cell error rate for random channel with finite buffer.....	86
3.10	ATM node model .....	89
3.11	Markov Chain model of one source .....	89
3.12	Flow chart for Simulation the code performance with bursty traffic .....	93
3.13	Effect of source burstiness on cell loss rate .....	96
3.14	Simulation of bursty source with clustering coefficient 1.0.....	97
3.15	Effect of traffic intensity on cell loss rate under bursty traffic.....	99
3.16	Effect of buffer size on cell loss rate under bursty traffic .....	101
3.17	Effect of buffer size on cell loss rate under bursty traffic .....	102
3.18	Cell loss rate versus buffer size for bursty traffic .....	104
3.19	Cell loss rate versus buffer size for random traffic .....	104
3.20	Cell loss rate versus clustering coefficient for random channel.....	106
3.21	Effect of traffic burstiness on cell loss rate for correlated channel .....	108
3.22	Effect of traffic intensity on cell loss rate for correlated channel .....	110
3.23	Effect of buffer size on cell loss rate for correlated channel.....	112
3.23	Effect of buffer size on cell loss rate for correlated channel.....	113
3.24	Comparison between random and correlated channel performance under bursty traffic ... .....	115

## LIST OF SYMBOLS

AAL	:	ATM Adaptation Layer
ARQ	:	Automatic Repeat Request
ATM	:	Asynchronous Transfer Mode
B	:	Buffer Size
B-ISDN	:	Broadband Integrated Services Digital Network
BER	:	Bit Error Rate
$c$	:	Clustering Coefficient
CER	:	Cell Error Rate
CLR	:	Cell Loss Rate
CRC	:	Cyclic Redundancy Code
$f$	:	Buffer overflow probability
FEC	:	Forward Error Correction
HEC	:	Header Error Control
ISDN	:	Integrated Service Digital Networks
QoS	:	Quality of Service
RS	:	Reed Solomon Codes
$\rho$	:	Traffic Intensity
$\omega$	:	Correlation Coefficient

## ملخص الرسالة

اسم الطالب: محفوظ صالح بن محفوظ  
عنوان الرسالة: دراسة كفاءة أنظمة التحكم بالأخطاء في شبكات ATM  
التخصص: الهندسة الكهربائية  
تاريخ التخرج: صفر ١٤٢٢ هـ

شبكات البث المكثف السريعة و الاتصالات المتنقلة اللاسلكية قوتان قائدتان رئيستان في صناعة تكنولوجيا الاتصالات حاليًا. إن تقنية طور النقل غير المترامن ATM تعتبر من أحد التقنيات التي تعد بتوفير تطبيقات عالية السرعة في مجال نقل المعلومات. تعاني شبكات ATM من فقدان الخلايا عند الإرسال عبر القنوات المشوشة، مثل القنوات اللاسلكية، بالإضافة إلى فقدان الخلايا بسبب فيضان وحدة التخزين (Buffer overflow). لمتل هذه المواقف، أنظمة التحكم بالأخطاء يجب أن تدرج لاسترداد الخلايا المفقودة. هذا البحث قام باستخدام شفرة ثنائية الأبعاد لاستعادة الخلايا المفقودة في شبكات ATM. يتميز هذا النظام ببساطة تطبيقه على الخلايا و بمحاظفة شكل خلايا ATM. لقد تم دراسة أداء و كفاءة هذا النظام للتحكم بالأخطاء ضمن القنوات ذات التشويش العشوائي وذات التشويش المترابط. كذلك تمت معاينته عند حدوث فيضان وحدة التخزين عند مرور الأحمال العشوائية و المترابطة.

درجة الماجستير في العلوم  
جامعة الملك فهد للبترول والمعادن  
الظهران- المملكة العربية السعودية

# ABSTRACT

**Name:** MAHFOOZ SALEH BIN MAHFOOZ

**Title:** RELIABILITY ANALYSIS OF ATM NETWORK  
WITH ERROR CONTROL

**Major Field:** Electrical Engineering

**Date of Degree:** May 2001

*High-speed Broadband networks and Wireless mobile communications are presently the two major driving forces in the telecommunication industry. Asynchronous Transfer Mode (ATM) has emerged as a promising high-speed transport technology for multi-service applications in broadband networks. The transmission of ATM over noisy links, such as wireless, suffers from transmission errors which cause cells to be lost. Cell loss may occur also due to buffer overflow. For such situations, error control schemes must be incorporated to recover lost cells. In this case, two-dimensional matrix coding scheme for operation over noisy ATM is used. This coding scheme is simple to encode and decode (modulo-2 addition) and it preserves the ATM cell format. The Performance of the code is investigated over noisy links considering the effect of random and correlated channel errors, and that of buffer overflow under random and bursty traffic loading.*

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

# **CHAPTER 1**

## **INTRODUCTION AND LITERATURE REVIEW**

High-speed Broadband networks and Wireless mobile communications are presently the two major driving forces in the telecommunication industry. Asynchronous Transfer Mode (ATM) technology has been adopted as the platform for Broadband Integrated Service Digital Networks [1]. ATM has been designed to be used over high quality transmission links, such as optical fiber, characterized by high data speed and low bit error rates. The ATM employs only header error control to protect ATM cell header from bit errors. Since ATM specification have been developed for high quality optical fiber transmission stream, header error control was sufficient for desired performance in ATM networks. There has been great interest lately in the utilization of ATM over wireless

links. The transmission of ATM cells over noisy links requires powerful correcting mechanism to protect the ATM cell, which includes protection of the header to reduce the cell loss rate and protection of the payload to make the noisy link more reliable.

Since the research in this thesis falls in the area of error control in ATM networks, this chapter provides an overview of these two subjects. Some important concepts of ATM networks are summarized in section 1.1. To keep it short, the aspects discussed for ATM networks are those related to our study in this thesis. A general overview of channel coding and error control techniques is given in section 1.2. An up-to-date literature survey of error coding schemes for ATM network is provided in section 1.3. A detailed statement of the problem of this work is furnished in section 1.4.

### **1.1 ATM Networks:**

Asynchronous Transfer Mode (ATM) is a new communications technology, which is fundamentally and radically different from previous technologies. Its commercial availability marks the beginning of what promises to be a genuine revolution in both the data communications and telecommunications industries.

Around 1988 the telecommunications (carrier) industry began to develop a concept called Broadband Integrated Services Digital Network (B-ISDN). This was conceived as a carrier service to provide high-speed communications to end-users in an integrated way.



ATM is the switching technique recommended by International Consultative Committee for Telecommunication and Telegraphy (CCITT), to carry B-ISDN traffic [2].

### **1.1.1 ATM Technology:**

ATM is a cell-based switching and multiplexing technology designed to be a general-purpose, connection-oriented transfer mode for a wide range of services. ATM is also known as cell relay. The word asynchronous in ATM comes from the fact that time slots are not assigned to specific users as it is the case with synchronous transfer mode (STM); instead time slots are available to any user who is ready to transmit data [3].

ATM differs from other technologies such as packet switching and frame relay in providing simple routing, guaranteed switching delays and light protocols.

In ATM networks, all information is formatted into fixed-length cells. The fixed cell size ensures that time-critical information such as voice or video is not adversely affected by long data frames or packets. At the edges of the network, user data frames are broken up into cells. At the destination side of the network, the user data frames are reconstructed from the received cells and returned to the end user in the form (data frames, etc.) that they were delivered to the network. This adaptation function is considered part of the network but is a higher-layer function from the transport of cells.

### **1.1.2 ATM Characteristics:**

ATM has basic characteristics that distinguish it from other switching techniques [4].

- No error protection or flow control on a link-by-link basis:

Since optical links of an ATM network have a very low bit error rate, no action is taken when an error occurs during transmission. Moreover no flow control functions are being considered and end-to-end protocols are employed to correct transmission errors.

- Connection oriented mode of operation:

A connection setup is needed before any information transfer. Resources are reserved during the setup phase and the connection is rejected if no sufficient resources are available. Upon call completion, network resources are released. With this mode of operation minimal cell loss rate is guaranteed. This mainly simplifies cell routing and minimizes cell header information.

- Reduced header functionality:

The information contained in the cell header is very limited. Basically it contains the identification of a connection and its route. This guarantees fast processing and consequently short delays between communicating parties.

- Relatively small information field:

The small fixed-size information field (48 octets) has advantage of reducing the buffering requirements and hence reducing the queuing delays.

### **1.1.3 ATM Protocol Hierarchy:**

ATM networks have layered architectures. However, the functional layering in the B-ISDN protocol model does not follow the international standard Open System Interconnection (OSI) model [4]. The B-ISDN/ATM hierarchy consists of the physical medium-dependent layer (PMD), the ATM layer, the ATM adaptation layer (AAL), and the higher service layer(s).

The PMD layer is responsible for the proper bit transmission and performs functions which are necessary to insert/extract the cell flow into/out-of a transmission frame.

The ATM layer contains all the details of the ATM technique, and is common to all services. This layer is physical medium independent. The data unit of this layer is the cell and the ATM layer performs the function of delivering cells from the source to the destination. This layer also performs the cell-based multiplexing/demultiplexing and cell delineation.

The AAL and higher services layers of the ATM protocol model are service-dependent. The boundary between the ATM layer and the AAL corresponds to the difference between functions applied to the cell header and functions applied to the information field. AAL provides an interface between user and the ATM layer for applications with similar services requirement. AAL is not part of user plane at the intermediate ATM switches inside the network. Different AALs are defined to support different types of traffic.

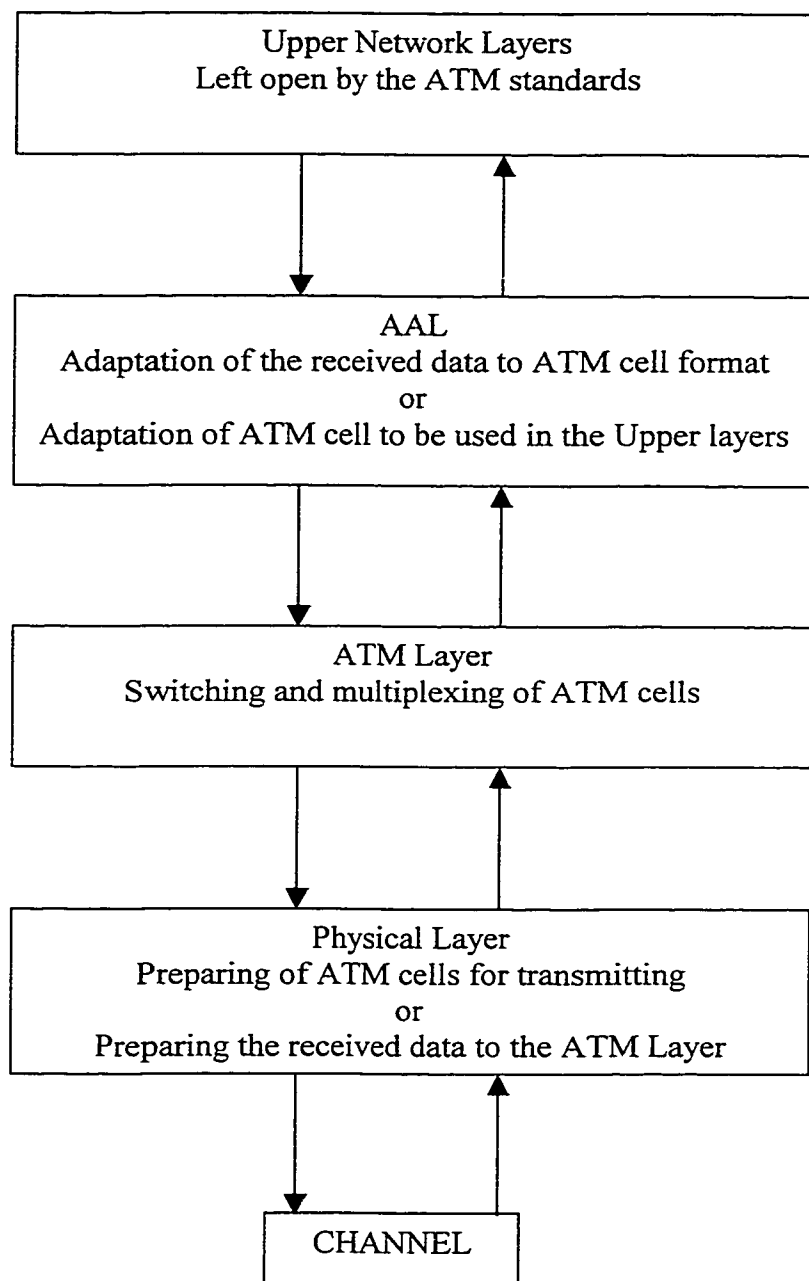


Figure 1.1: ATM Network Layers

### 1.1.4 ATM Cell Structure:

The structure of the cell is important for the overall functionality of the ATM network [5]. A large cell gives a better payload to overhead ratio, but at the expense of longer, more variable delays. Shorter packets overcome this problem, however the amount of information carried per packet is reduced. A compromise between these two conflicting requirements was reached, and a standard cell format chosen. The ATM cell consists of 53 bytes. Out of these there are 5 bytes that constitute the header, and 48 bytes that constitute the payload, as shown in Figure 1.2.

The information contained in the header is dependent on whether the cell is carrying information from the user network to the first ATM public exchange (User-Network Interface UNI), or between ATM exchanges in the trunk network (Network-Node Interface – NNI). They are similar except for the first byte. Both types are shown in Figure 1.3.

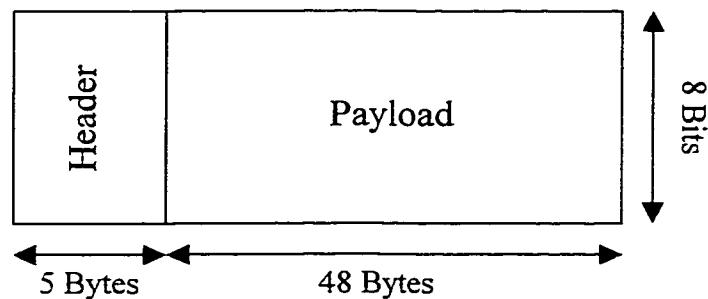


Figure 1.2: ATM Cell Format

The ATM cell header illustrated in Figure 1.3 consists of five fields: virtual path identifier (VPI), virtual channel identifier (VCI), payload type (PT), cell loss priority (CLP), and header error check (HEC). At a demarcation point between an ATM end station and the network, the cell header also includes a generic flow control (GFC) field. This 4-bit field is part of VPI in the network [6].

GFC provides a framework for flow control and fairness to the user-to-network traffic and does not control the traffic in the other direction (i.e., network-to-user traffic flow).

In ATM, end-to-end virtual channels are established between end stations before the traffic can start flowing. Routing of cells in the network is performed at every switch for each arriving cell. The routing information of a cell is included in the two routing fields of the header: VPI and VCI. The two levels of routing hierarchies, virtual paths (VP) and virtual channels (VC), are defined as follows:

- VC: A concept used to describe unidirectional transport of ATM cells associated by a common unique identifier value, referred to as VCI.
- VP: A concept used to describe the unidirectional transport of cells belonging to VCs that are associated by a common identifier value, referred to as VPI.

The PT is used to specify whether the contents of the payload consists user data or management data. Also the PT is used for including the congestion notification bits.

The CLP is one bit field used for cell loss priority. Cells with CLP set (low priority) may be discarded earlier at congested switches than cells with CLP bit not set (high priority). One example would be if voice is encoded in to equal numbers of high and low

order bits and sent over cells that alternately carries high and low order bits. If the CLP field of the low order bit cells are assigned by 1, they will be the first cells to discard during congestion hoping for minimal degradation in voice quality [7].

The HEC field is used mainly for two purposes: for discarding cells with corrupted headers and for cell delineation. CRC-8 code parity is used in the HEC field to protect the valuable information of the header, which is used in one of two modes: pure error detection or single error correction and multiple error detection. When errors are detected (and can not be corrected), the cell is discarded.

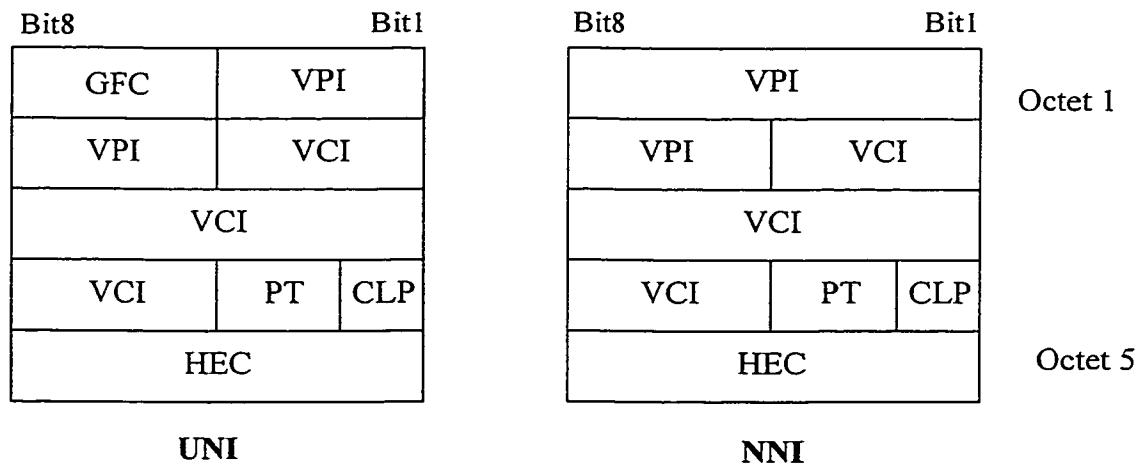


Figure 1.3: Structure of the Header of ATM cells for both the interfaces.

### **1.1.5 ATM Switching:**

The simplest form of an ATM switch is illustrated in Figure 1.4. A number of communication links are connected with some receiving cells and others transmitting them. These links, while logically separate, are paired such that one inbound and one outbound connection are joined to form a physical full-duplex link. Cells are received by the switch and retransmitted on one of the outbound links according to the routing rules of ATM.

The core of the switching process is as follows [2]:

1. A cell is received on an inbound link and its header is examined to determine on which outbound link it must be forwarded.
2. The VPI/VCI fields are changed to new values appropriate to the outbound link.
3. The cell is retransmitted towards its destination on an outbound link.
4. During this process the system is constrained to deliver cells to the appropriate output port in the same sequence as they arrive. This applies to each VC (there is no requirement to retain the sequence of cells on a VP provided that on each VC the cells remain in sequence).

If the ATM switch is nonblocking, there will be a path from every input to every output port [8]. Even if the switch is nonblocking internally, contention for output port could take place since cells arriving at two or more inputs may try to go to the same output simultaneously. If the time that the contention persists is short, contention can be handled



by buffers. The position of buffers within a switch greatly influences the switch performance, and at a high level of abstraction three possible disciplines can be used:

- Input buffering
- Output buffering
- Shared buffering

A combination of these three variants can also be considered to optimize the implementation cost and performance.

#### **1.1.5.1 Input Buffering:**

Buffers at the input port (Figure 1.5(a)) may be used against both internal blocking and output contention. In general, an input buffer is implemented as a First In First Out (FIFO). A cell that cannot be switched to its output port during the current cycle occupies the Head of Line (HOL) position at the FIFO and retries during the consecutive cycles. In this case, cells behind the HOL cell are forced to wait in the FIFO until they reach the HOL position in the queue. In particular, during a cycle, a path from an input port to the output port of non-HOL cells waiting in the queue can be established without causing any internal or output blocking of any HOL cell at other ports. However, input buffering does not allow such cells to be switched during the current cycle, thereby forcing some internal links to be idle even though there are cells waiting in the system that can use these links.

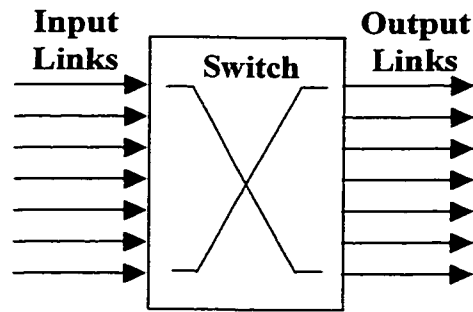


Figure 1.4: ATM Switch schematic

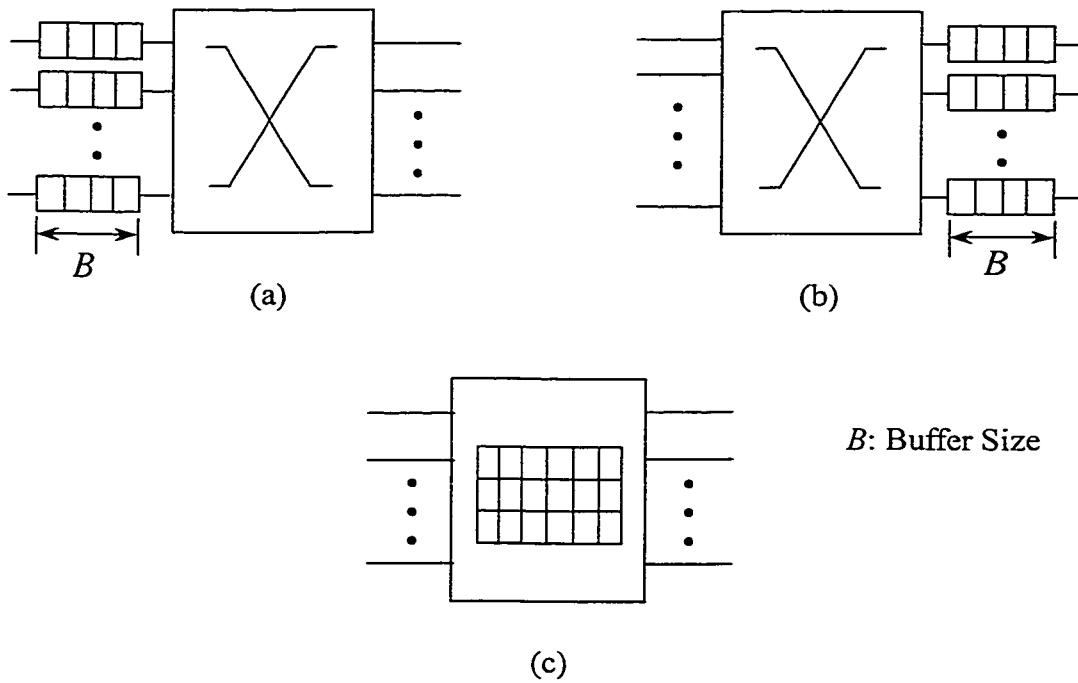


Figure 1.5: Buffering strategies of ATM switches: (a) Input buffering; (b) Output buffering; (c) Shared buffering

### **1.1.5.2 Output Buffering:**

Due to output contention, an internally nonblocking switch may still block at the output ports. If output buffering is not used, only one of the contending cells can go through the output port. Depending on the scheme used, other contending cells may be dropped (i.e., lost) or stored at either input or internal buffers. With output buffering (Figure 1.5(b)), all cells contending for the same output port are stored at the output ports until they can be read out by the corresponding transmission link. Hence, output buffering results in a better switch throughput than input buffering, since only one of the contending cells from different input ports can be delivered to the output in the latter. The output buffering is optimal in terms of throughput and delays, but it needs some means of delivering multiple cells per cell time to any output. Hence, either the output buffers must operate at some factor times the port speed, or there should be multiple buffers at each output. In both cases, the throughput and scalability are limited, either by the speedup factor or by the number of buffers.

### **1.1.5.3 Shared Buffering:**

In the shared buffering (Figure 1.5(c)) the queueing buffers are not dedicated to a single input or output, but shared between all inputs and outputs. The shared buffering behaves exactly as the output buffering. However, since multiple queues are combined in a single physical memory, a major advantage of the shared buffering is reflected in the number of

cells to be stored in the buffer, but more complicated control logic is required to ensure that the FIFO discipline for each output link is performed in the single shared memory.

#### **1.1.6 Congestion in ATM:**

Symptom congestion is when the number of cells within the network causes the performance to fall off dramatically [9]. Specific to ATM, congestion occurs when offered load from the user to the network approaches or exceeds the network design limit for guaranteeing the Quality of Service (QoS) agreed upon. The sources of congestion are limited resources such as buffer size, and inherent characteristics of switching nodes such as multiplexing delay. Generally in a congested switch, low priority cells are rejected once the switch buffer is full. Priority cells are rejected when there are no low priority cells to discard from the queue (buffer). Priority cells may contain critical data such as control information, so losing such cells could affect the QoS provided.

#### **1.1.7 Wireless ATM**

The growth of wireless communications paired with the rapid developments in asynchronous transfer mode (ATM) networking technology signals the start of a new era in telecommunications. The growth of cellular radio communications in the past decade has been remarkable. The number of cellular users has exceeded all predictions on

cellular's use. Demand for cellular communications has placed a heavy demand on the capacity of wireless/air interfaces and the network resources available. The success of cellular mobile communications has spurred the telecommunications industry to push the implementation of Personal Communications Services (PCS). PCS will provide voice, text, video and data. As a result, the demand for higher transmission speed and mobility is even greater.

A wireless ATM network, which is designed to provide high-speed isochronous and asynchronous communications for wireless users, is a good match for these demands.

The introduction of ATM in wireless environment creates many interesting challenges including how to deal with wireless links at high and variable error rates. The problem arises as ATM was designed for very low error rate environment and hence strong error control was not provided. For example, only the header of ATM is protected by a HEC field, capable of detecting single and multiple bit errors in the header. There are several ways to counteract the errors caused by wireless links such as the use of equalization, diversity, error detection and retransmission, error correction, error correction and retransmission (also called hybrid ARQ), and error resilience.

Equalization is used to combat Inter-symbol interference (ISI) that occurs due to multipath propagation of signal from transmitter to receiver over wireless link, thus it compensates for the average range of expected channel amplitude and delay characteristics. It requires the use of periodic training sequence to effectively cancel ISI.

It is used in systems where data is transmitted in a short time blocks (TDMA is well suited for equalizers).

Diversity is a powerful communication technique to compensate for channel impairments. It can be of antenna diversity, time diversity or frequency diversity. Interleaving that convert burst errors to random errors for transmission over burst error channel is considered a time diversity technique.

One way to counteract channel errors is to allow receiver to detect and/or correct errors by sending redundant bits. This is also termed as error control coding, and it will be discussed in the next section.

Error resilience, which is the ability to deal with errors that can not be corrected for some reasons, can be achieved in data transmitted over wireless links by using specialized procedures. These include error concealment, temporal localization, and spatial localization. One example is MPEG-2 standard, where elaborate procedures are developed for achieving error resilience.

## **1.2 Channel Coding**

It is known that Bit Error Rate ( $BER$ ) is a function of the signal to noise ratio ( $E_b/N_o$ ) and the modulation scheme used. The desired level of accuracy in the information delivered, i.e. desired  $BER$ , can be achieved by supplying enough power in the transmitted signal or by the use of more powerful modulation scheme to overcome

channel disturbance that produce errors. An alternative to increasing signal power is to use error control coding in which the same *BER* can be achieved with less transmitted power or the same power can produce a better *BER*.

The utility of coding was demonstrated by the work of Shannon [10]. In 1948, Shannon proved that if the data source rate is less than a quantity called channel capacity, communication over noisy channel with an error probability as small as desired is possible with proper encoding and decoding.

Channel coding is used to improve the reliability of data delivered to the user. The channel encoder accepts information bits from the source and adds redundancy bits to it to form a codeword that is transmitted through the channel.

The redundancy added by the channel coding is used for *Error detection* and/or *Error correction*. In some cases this redundancy may be used in different way for *Erasers Decoding*. These techniques are explained briefly in the following subsections.

### **1.2.1 Error correction and detection:**

The received coded data may contain some errors due to channel imperfections. In the receiver, the decoder tries to identify the invalid codewords. Any valid codeword is assumed to be correct. This process is called error detection. The decoder will fail in error detection if and only if the errors cause the codeword to become another valid codeword. On the other hand in error correction the decoder may attempt to correct some of the

received words. Error detection and error correction are achieved using the redundancy added by the encoder. The redundancy bits are added according to the rules of the used code. Any block code is characterized by the Hamming distance  $d$ . A code with Hamming distance  $d$  can detect up to  $d-1$  errors when used for error detection only, or correct up to  $\left\lfloor \frac{d-1}{2} \right\rfloor$  errors when used for error correction only. In general, a code with minimum distance  $d$  can correct up to  $t$  errors and simultaneously detect up to  $\lambda$  errors such that  $d = t + \lambda + 1$  [11].

### **1.2.2 Error Control Coding:**

There are two error control strategies have been popular in practice, namely Forward Error Correction (*FEC*) and Automatic Repeat Request (*ARQ*). In the ARQ, if no errors are detected by the decoder, the original information bits are delivered to the user. Otherwise, the receiver discards the received word and sends back a negative acknowledgment (*NACK*) requesting for a retransmission of the same codeword. Theoretically this process is repeated until the codeword is successfully received. Obviously, in any ARQ system the availability of a feedback channel is a must. From the above, it is clear that coding is used for the sole purpose of error detection. That is why high rate error detecting codes are used in ARQ systems.

There are three basic protocols of retransmission in an ARQ system: stop-and-wait (*SW*), go-back-N (*GBN*) and selective-repeat (*SR*). In SW-ARQ the transmitter stops



after each transmission and remains idle waiting for acknowledgment (ACK) to be received back. If ACK is received, a new cell is transmitted, while NACK means to re-transmit the same cell. In GBN-ARQ and SR-ARQ the transmitter continuously transmits cells until it receives NACK. In the case of GBN-ARQ the system re-transmits the negatively acknowledged cell and successive cells as well. In SR\_ARQ, only the NACK cells are retransmitted.

In FEC systems the redundancy added by the channel encoder is used by the decoder for error correction of the detected invalid code word. When the decoder detects the presence of errors in a received word it attempts to locate and correct them, and delivers the decoded word to the user. The number of overhead parity check bits needed for error correction is much larger than those needed for error detection. Also number of parity bits is increased as the number of errors that the code could correct increase. However, as more and more parity check bits are added, the required transmission bandwidth goes up as well. Because of the resultant increase in bandwidth, more noise is introduced and the probability of error increases. Also, as the code length is incremented it becomes increasingly more difficult and expensive to design good decoders. Therefore the goal is to choose parity bits to correct as many errors as possible, while keeping the communications efficiency as high as possible [10], [12].

The performance of digital communication systems is usually evaluated by their “throughput” and “reliability”. Throughput is known by the ratio of useful information bits accepted by the receiver to the total transmitted bits per unit time. Reliability is a

measure of correctness of the received data. Both error control strategies enhance the reliability of communication system at the cost of the throughput.

In FEC the decoded word must be delivered to the user regardless of whether it is correct or not. This implies that the reliability of systems using FEC technique is less than those using ARQ systems especially for poor channels. As matter of fact, ARQ can give almost the same reliability as that of an ideal channel. The only degradation of reliability from the ideal case comes from the possibility that an erroneous word passes undetected. The probability of this event is very low in relatively good channels especially when a strong error detecting code is used.

Since there are no retransmissions in a FEC system, it provides constant throughput efficiency, set by the code rate, regardless of the channel conditions. In an ARQ system, the process of retransmission degrades the throughput efficiency especially for poor channels.

In summary, both techniques have relative advantages and disadvantages. A combination system that has the advantages of both ARQ and FEC systems is termed in the literature *Hybrid ARQ*. In such systems an FEC is used in conjunction with ARQ to enhance the reliability with minimum degradation of the throughput.

In hybrid ARQ, an error detection code is used for ARQ and another code is used for FEC. At the receiver, the decoder first attempts to correct any error in the received codeword using the error correction code. Then the error detection code tries to detect any error in codeword received from the previous stage. If the decoder fails to detect an

error, it assumes that it is a correct codeword and it is delivered to the user. Otherwise if an error is detected the receiver discards the code word and requests a retransmission of the same codeword.

There are many types of codes that can be used in channel coding. Examples of the most famous types of codes are parity check codes, Hamming codes, CRC codes, Bose-Chaudhuri-Hocquenghen (BCH) codes and Reed-Solomon (RS) codes. Those codes are all from one family called block codes. Another family of codes is convolutional codes. Component codes can be used to enhance the error correction capability as in product codes, concatenated codes and turbo codes. Most of the codes can be used for error detection and/or error correction.

In the following subsections some of coding schemes will be explained that will be used in our work.

#### **1.2.2.1 CRC codes:**

The Cyclic Redundancy Codes (CRC) are extremely well studied and widely used for error detection only [13], [14]. That is why they are some times defined as high speed error detecting codes [14]. CRC codes are very popular due their powerful capability and the simplicity of implementing encoders and decoders. CRC is widely used in computer

industry [11]. The following is a list of famous CRC codes with their generator polynomials [14]:

<b>CRC CODE</b>	<b>GENERATOR POLYNOMIAL</b>
CRC-4	$g_4(x) = x^4 + x^3 + x^2 + x + 1$
CRC-7	$g_7(x) = x^7 + x^6 + x^4 + 1$
CRC-8	$g_8(x) = (x^5 + x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1)(x + 1)$
CRC-12	$g_{12}(x) = (x^{11} + x^2 + 1)(x + 1)$
CRC-ANSI	$g_{ANSI}(x) = (x^{15} + x + 1)(x + 1)$
CRC-CCITT	$g_{CCITT}(x) = x^{16} + x^{12} + x^5 + 1$
CRC-SDLC	$g_{SDLC}(x) = x^{16} + x^{15} + x^{13} + x^7 + x^4 + x^2 + x + 1$
CRC-24	$g_{24}(x) = x^{24} + x^{23} + x^{14} + x^{12} + x^8 + 1$
CRC-32A	$g_{32_A}(x) = x^{32} + x^{30} + x^{22} + x^{15} + x^{12} + x^{11} + x^7 + x^6 + x^5 + x$
CRC-32B	$g_{32_B}(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

When an  $(n, k)$  CRC code is used for error detection only, they are powerful enough to detect all the following types of errors [13],[14]:

All combinations of  $d_{\min} - 1$  random errors or less.

All errors of length  $\leq (n - k)$ .

$1 - 2^{1+k-n}$  of error bursts of length  $n - k + 1$

$1 - 2^{k-n}$  of error bursts of length  $> n - k + 1$

All the error patterns of an odd number of errors, provided that  $g(x)$  has an even number of nonzero coefficients.

### **1.2.2.2 Parity-Check Codes:**

In single parity check codes, a single parity bit, or check bit, is appended to a block of  $k$  information bits, the check bit being chosen to satisfy an overall parity rule for the codeword. Encoding of the single parity check code is described by the equation

$$i_1 + i_2 + \cdots + i_k + p_1 = 0 \quad \text{mod } 2$$

where  $i_1, i_2, \dots, i_k$  are arbitrary information bits, and  $p_1$  is the parity bit. The equation specifies that the parity bit be chosen so that the codeword have even parity, that is an even number of ones. Odd parity codeword is obtained by letting the modulo-2 sum of the equation equal to one. Most, though not all, coded systems in operation today use the even-parity rule [12]. The even parity check codes are error-detecting codes; they are able to detect any odd number of errors.

### **1.2.2.3 Product Codes:**

A product code is a code which uses some simple codes to build a relatively more complex and powerful code. Elias (1954) introduced the idea of the product code, constructing a two-dimensional array from two given component codes [15]. Product codes are achieved by arranging the information bits in a two-dimensional array. Then rows are encoded using some code ( $C_1$ ) and then the columns are encoded using code ( $C_2$ ). The codes  $C_1$  and  $C_2$  are block codes, which may be the same code or different

codes. In general, product codes are much easier to decode than non-product codes with same size and rate. Figure 1.6 gives the general shape of the code. In general, each block unit in the matrix may represent a bit or an array of bits. Product codes are capable of correcting random errors as well as burst errors [15]. The simplest example of a product code is that where  $C_1$  &  $C_2$  are single-parity check codes. Such code has a Hamming distance of  $d=4$  and thus capable of single error correction. When used for erasure decoding it is capable of recovering any pattern of three or less erasures, and many patterns of larger arrangements.

The code under study in this work is a simple single parity check code that is applied to rows and columns. Applying code to ATM networks, each block unit of the code is taken to be an ATM cell. The code is constructed by arranging cells in a matrix. Those cells are encoded by appending a parity check cell to each row and each column. The parity check cell at the end of each row is obtained by bit-by-bit, mod-2 addition of the cells in that row, while the parity check at the end of the column is obtained by mod-2 addition of the cells in that column. Note that only the information field of an ATM cell is encoded while the same header is applied to the parity cell. A simple example with small packet size is presented in Figure 1.7. Figure 1.7 shows the code for a matrix size  $5 \times 5$  and cells of 5-bit long.

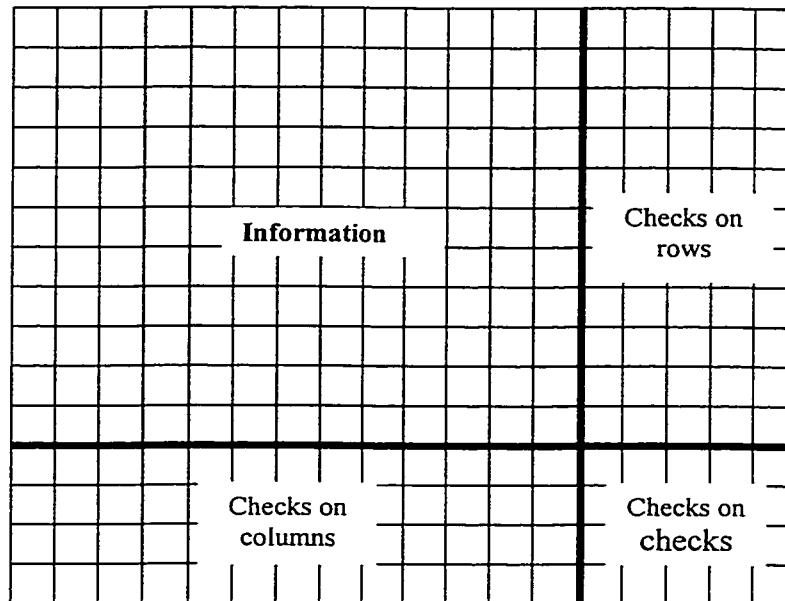


Figure 1.6: General structure of two-dimensional product codes.

11011	10001	00011	10101	00110	111010
01010	01110	10111	11001	00010	01000
11110	01001	11111	10011	10000	01011
10011	10101	00110	01011	10110	11101
10000	10010	10001	01101	10111	01001
01100	10001	11100	11001	10101	01111

Figure 1.7: Example of a single parity product code.

Decoding of the two-dimensional product code for recovery of lost cells is fairly simple in the area of erasure decoding described in section 1.3.3. The decoder starts by scanning the first row. If a single cell is lost it will be recovered by mod-2 addition of all cells in that row. If more than one cell is lost, the decoder will not perform any recovery job. The decoder continues scanning all the successive rows and acts in the same way described above. The same decoding procedure is then carried out for all columns.

### **1.2.3 Erasure Decoding:**

In the above discussion we have assumed that the channel has introduced errors in the received words; that is some bits have been inverted. There are cases where some of the transmitted bits or complete packets are missing or erased from the receiving data. In these cases the code must perform erasure decoding.

Erasures can take place for different reasons. For example in soft decision decoding, erasures are introduced when the received energy level is too low for the decoder to make reliable decision; hence the bit is erased. Decoders, sometimes, intentionally introduce eraser to simplify decoding. An example is found in some decoding algorithms of product code. In some network protocols complete packets may be discarded, lost or erased. In ATM networks an error in the header causes the cell to be discarded, also cell may be discarded due to buffer overflow.



For any code, its ability to fill erasures is double its ability to correct errors [16]. Therefore, a code with Hamming distance  $d$  can fill  $d-1$  erasures if used for erasure decoding only. If the code is used for erasure filling and error correcting simultaneously, then it can fill  $s$  erasures and correct  $e$  errors such that  $d = 2e + s + 1$  [10].

### **1.3 Literature Review:**

The growing interest in wireless communication services has pushed the research community to consider the use of the ATM technique into wireless access networks [17]. To make wireless links fit into ATM world some error control methods must be applied. Several methods were proposed in different papers to overcome this problem using different FEC techniques, which include convolutional codes, applying FEC to header and other schemes [18-25]. For non-delay sensitive applications the use of hybrid ARQ/FEC schemes was proposed [26-29]. The use of matrix code is also proposed for recover of lost cells [30-35].

Among the powerful FEC techniques are Convolutional codes. They were proposed to be used with ATM cells either alone, associated with fragmentation and puncturing [18], or concatenated with RS codes [19,20], with the use of higher layer ARQ to recover lost cells.

Some papers have proposed schemes in which forward error correction is applied to the header only [21,22] to protect cell from being lost due to channel errors. A powerful BCH code is applied to the header instead of HEC to overcome random errors and, thus, make the cell discard on the wireless links less frequent.

The use of separate codes for header and payload has also been proposed [23,25]. Rice and Moore [23] proposed an FEC scheme with two different codes. One code is applied to a compressed header used over mobile wireless links as suggested by [24]. The other code, variable rates code scheme based on the concept of multiple shortened code, is used to protect payload. Variable rate coding schemes are used to provide different error protections for different QoS parameters using the same code generator.

Aikawa et al. [25] also proposed an FEC scheme with two different codes that modify the ATM cell format. One code is a powerful BCH code used for the header, in place of HEC, to overcome cell loss. The other code is a higher rate BCH code applied to the payload to improve the BER or reliability sufficiently. The performance of the coding scheme was analyzed and compared to the HEC performance.

In case of non-delay sensitive applications the use of hybrid ARQ/FEC have been proposed in [26-28]. Chiani and Volta [26] investigated ARQ techniques and FEC codes for connection-oriented, non-delay critical and variable bit rate services. An adaptive error control scheme was proposed that chooses the control techniques for each

connection according to its requirements described by the connection. Two new specific layers have been introduced: the Wireless Physical layer (W-PHY) and the Wireless Data Link Layer (W-DLL). The network is structured in Pico-cells with a base station at the center of each cell. Three types of ARQ protocols were analyzed: selective repeat ARQ, selective repeat Type I Hybrid ARQ/FEC, and selective repeat Type II Hybrid ARQ/FEC. Infinite length of receiver buffer was assumed with noiseless feedback channel.

Andreadis et al. [27] proposed a new system for the interconnection of two ATM nodes via a wireless link for the transmission of data and multimedia services. The ATM cell format and length are modified in the wireless link, but the structure is maintained 'compatible' with ATM to simplify conversion procedure between the base station and sub-network station.

Jung et al. [28] suggested several error control methods to overcome the performance degradation in wireless links caused by burst errors. They provided performance comparison between several FEC and ARQ schemes through the use of packet error rate and throughput. FEC schemes were considered for delay sensitive traffic while ARQ were considered for non-delay sensitive traffic. A concatenated FEC scheme with inner convolutional code and outer non-binary BCH code was compared to an FEC scheme with two different codes for payload and header.

Borgonovo and Capone [29] investigated the problem of error control at the access interface of a wireless ATM in a channel with bursty errors and tight service-delay constraints. The efficiencies of three different error control schemes are compared, namely, FEC BCH code with interleaving, a real time ARQ scheme, and block erasure codes. It was found that delay constraints could severely limit the performance of the FEC code BCH with respect to other approaches.

The use of matrix code was proposed in [30-35], in which one dimension coding [30-32] and two-dimensional coding [33-35] used to recover cells that have been lost due to channel impairments or buffer overflow. Ohta and Kitami [30] proposed node-to-node and end-to-end FEC approaches. In the node-to-node approach, losses were detected at virtual path (VP) terminating nodes. The issue of lost cell detection was also addressed. In the end-to-end approach, sequence number (SN) is used to detect lost cells. While in node-to-node approach, virtual channel identifiers (VCI) in addition to the SN is used to form cell recognition patterns (CRP), which are used to detect lost cells. Data cells are gathered in a matrix and each row is terminated with a cell loss detection cell (CLD) to housekeep the data cells in a row. Each column in the matrix is terminated with a parity cell using simple parity coding (i.e. modulo-2 addition). A single lost cell in a column can be detected and corrected. Therefore a burst of lost cells of length at most equal to row length can be handled successfully. Longer bursts produce double lost cells in a column, which cannot be recovered by the parity cell.

In [31,32] a detailed analysis of a hybrid ARQ/FEC cell loss recovery scheme that is applied to virtual circuits of a multi-hop ATM network is given. The coding scheme used is similar to that used by Ohta and Kitami [30]. Error detection was performed on each hop, while coding and decoding for recovery of lost cells was performed at end-to-end basis. Different network parameters were encountered in the analysis such as number of transit nodes, traffic intensity and ARQ packet length. Also cell error performance was analyzed and used as a measure for the reliability of recovered cells. It was shown that the cell recovery matrix was very efficient technique for recovering lost cells.

In [33] a major improvement of the simple FEC technique in [30-32] is made by extending the parity check for lost cells recovery to be two dimensional. That was done simply by placing one column at the right of the matrix as a row-wise parity check and one row at the bottom of the matrix as column-wise parity check. By this method more lost cells patterns can be corrected. The analysis of cell loss rate in [31] shows a clear improvement over the one-dimensional case for the same amount of redundancy.

In [34], a novel study about the structure of lost cell patterns in two-dimensional matrix code was presented. The work introduces a new concept of basic patterns, from which all cell loss patterns may be generated. The pattern is the possible arrangement of lost cells in a decoding matrix. For a given number of lost cells these basic patterns are classified to recoverable basic patterns and unrecoverable basic patterns. And using the unrecoverable basic patterns three upper bounds B-I, B-II and B-III was derived for the

post-decoding cell loss rate. Every bound is obtained by tightening of the previous bound. B-III was considered the tighter bound which gives the optimum post-decoding cell loss rate that can be achieved by the code. The performance of the cell loss recovery scheme over noisy ATM was analyzed, assuming cell loss is only due to error detection in the cell header. Also the effect of using ARQ to recover lost cells by requesting retransmission of those cells or the whole matrix was investigated.

Ayanoglu and Oguz [35] analyzed the performance of two-level FEC cell recovery scheme. Their approach is very similar to the approach devised by Ohta and Kitami [30] except they apply the code in two directions, i.e. to rows and columns, instead to columns only. Also they used RS codes instead of simple parity coding. They studied the code performance under bursty traffic loading and finite buffer size. Detailed performance calculation are discussed and compared with simulation results. This approach, although highly tolerable to cell loss, is computationally complex and costly.

#### **1.4 Proposed work:**

The single-parity matrix coding scheme found in [30],[31],[32], and its extension to two dimensions [33],[34], [35] is very attractive for at least three reasons:

1. It does not modify the format of the cell. Thus the data cells remain unaltered in this coding procedure. Also, the parity cells use the standard cell format.
2. It is extremely simple to encode and decode (modulo-2 addition)
3. The two-dimensional scheme is amazingly powerful in cell loss recovery

However, it is found that all of the above schemes have simplified the analysis by making some or all of the following assumptions:

- (a) Random channel errors
- (b) Infinite Buffer Size (no buffer overflow)
- (c) Finite buffer size
- (d) All recovered cells are correct

For example, [31],[32] made assumption (a) and (c), while [30] made the assumption (c). The two dimensional scheme in [34], [33] was analyzed under the assumptions (a), (b) and (d).

Motivated by its advantages, the two-dimension single-parity coding scheme, is adopted for this work. A detailed study of the performance of the code when used to recover lost cell over a noisy ATM link is carried out. We alleviate all the assumptions imposed in previous works in order to get a more accurate evaluation of the scheme under practical conditions.

In particular, we study the following effects

- (1) Random error channels as well as correlated error.
- (2) Finite buffer size. Both random traffic loading and bursty traffic loading are considered.
- (3) Cell error rate which provides measure for the reliability of the data at the output of the system.

The proposed work is organized in two chapters. The code performance is analyzed with infinite buffer size for both random and correlated channel in Chapter 2. The effect of finite buffer size on the code performance with random traffic loading and bursty traffic loading is studied in Chapter 3. We conclude with Chapter 4, where we summarize the findings of this work and suggest directions for future work.



# CHAPTER 2

## RELIABILITY STUDY: INFINITE BUFFER CASE

### 2.1 Introduction

In this chapter we analyze the two-dimensional matrix code under the infinite buffer case. We consider both random and correlated channels. The two-dimensional matrix code is applied to ATM traffic when it is transmitted over a noisy or poor ATM link, such as wireless link, as shown in Figure 2.1. It is assumed that the ATM traffic traverses single noisy ATM link.

The FEC code matrix of  $M \times N$  cells is formed by arranging the data cells in an  $(M-1) \times (N-1)$  matrix, then appending a parity cell to each row and column as depicted in Figure 2.2. The column (row) parity cells are simply generated by applying module-2 addition to all data cells in the column (row). The data cell and parity cell format is shown in Figure 2.3. It should be noted that the mod-2 addition, used to generate parity cells, is applied to the payload segments except for the field of sequence number SN. The SN field is used to detect any lost cell or wrongly inserted cell in a row or series of cells.

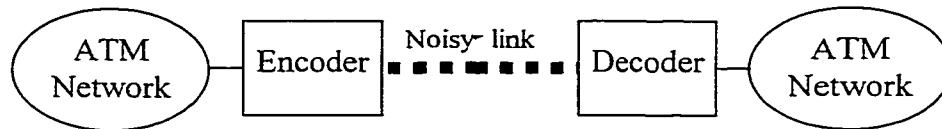


Figure 2.1: Configuration of the FEC model

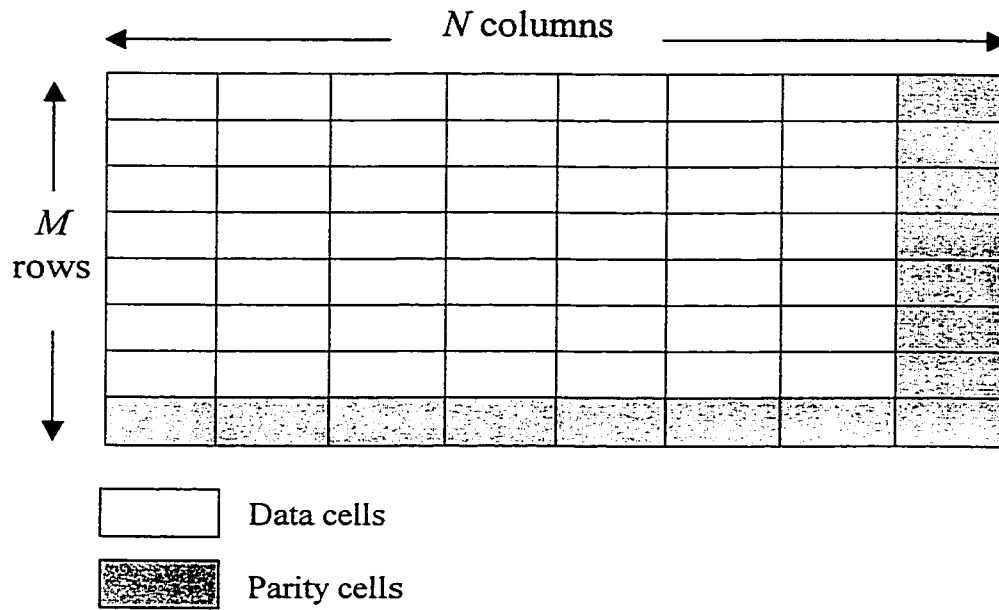


Figure 2.2: Format of the coding matrix

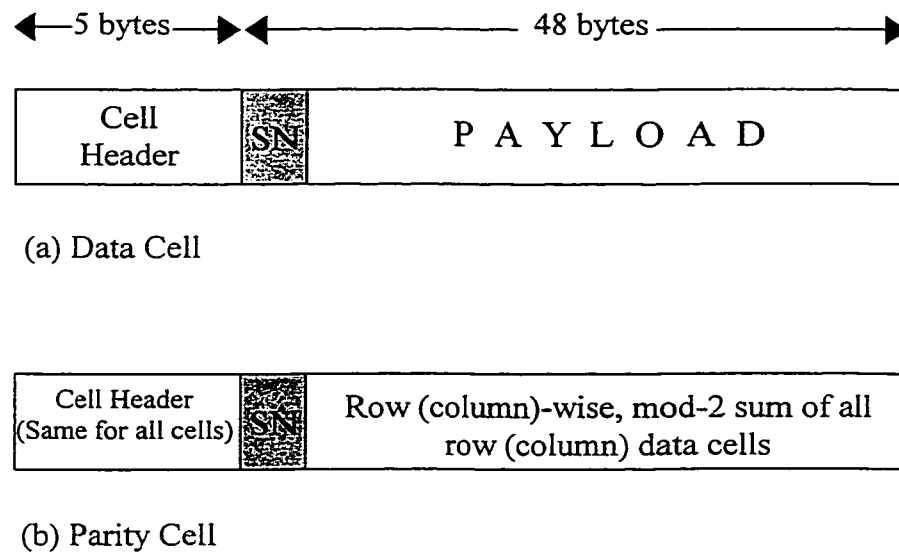


Figure 2.3: The Data and Parity cell formats

The data cells fill a coding matrix row-wise for encoding as being transmitted in parallel. Thus, with the simple mod-2 encoding used, the delay introduced by the encoder is very small. The encoder generates parity cell for every row, under the control of a peak rate policing mechanism, i.e. the number of cells will not exceed the guaranteed value of peak cell rate. Similarly, a parity cell is generated for each column and transmitted after the last row.

The decoding is performed at the receiving end, where the received cells are rearranged in an  $M \times N$  matrix format. It is assumed that the positions of missing cells are known and are replaced by dummy all-zero cells. Usually this is done by the use of special cell known as Cell Loss Detecting cell which are appended at the end of each row. This cell contains cell recognition patterns corresponding to the cells in the row. Using the cell recognition patterns and comparing them to the received cells the position and number of lost cell in a given row are found. It is assumed that this mechanism is applicable and it will not be considered in our work. Then the decoder performs column-wise and row-wise decoding to recover lost cells. When a single cell is lost in a row or column, it can be recovered by a simple parity checking (mod-2 addition). The process of lost cell regeneration is shown in Figure 2.4. If two or more cells are found lost in a row (column), that row (column) is skipped.

In general, many rounds of decoding may be carried out (row, column, row, ...) until all lost cells are recovered or no more recovery is possible. However it is found that [34] the majority of lost cells are recovered in two rounds. The insignificant improvement

in doing further rounds does not, for most of time, justify the increased delay in decoding. For this reason we limit the decoder to two rounds.

Two performance criteria are evaluated: the cell loss rate and cell error rate. Both random channels and correlated channels will be considered. We will study the effect of varying the channel bit error rate, the matrix size and the correlation coefficient, in case of correlated channel, on the two performance criteria.

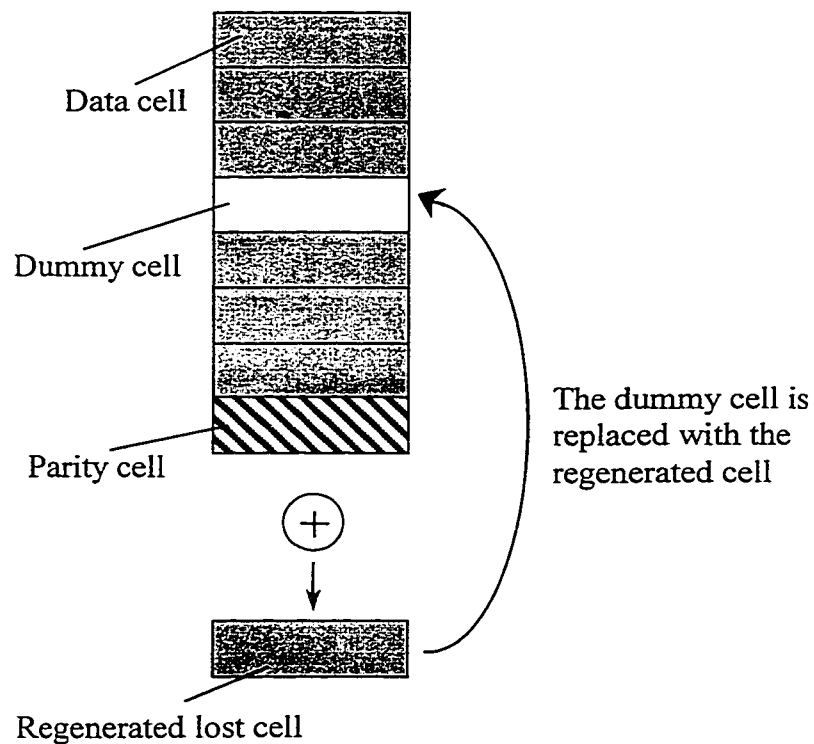


Figure 2.4: Lost cell regeneration

The performance of two-dimensional code is evaluated in the following for both random channel and correlated channel errors. The cell loss due to channel errors only will be considered in this chapter.

## **2.2 Random Channel Case:**

In the following we evaluate the performance of the code under random error channels.

### **2.2.1 Pre-decoding probabilities:**

At the ATM node, the header is checked for errors; if errors are detected the cell is discarded.

Denoting,

$p$  = bit error probability of the noisy link

$n$  = number of bits of payload

$n'$  = number of bits of the header

$m$  = cell size in bits =  $n + n'$

The probability  $P_c$  that the cell is received correctly, which takes place when both the header and payload segments survive the physical layer errors, is given as,

$$P_c = (1 - p)^m \quad (2.1)$$

On the other hand, a cell is received in error when the header is received correctly but the payload segment is affected by channel errors. In deriving 2.1 it was assumed that all error patterns in the header segment are detectable. Considering the power of CRC-8 code, which is used in HEC, in error detection, this assumption is well justified. Denote the probability of this event by  $P_e$ . Then,

$$P_e = (1 - p)^{n'} [1 - (1 - p)^n] \quad (2.2)$$

A cell is lost when the header is found in error, regardless of the state of the payload. Therefore, the probability  $P_l$  of cell loss is then given by

$$P_l = 1 - (1 - p)^{n'} \quad (2.3)$$

Note that,

$$P_l + P_e + P_c = 1$$

### **2.2.2 Post Decoding Probabilities:**

Next we evaluate the post-decoding probabilities. Let's start by considering the first round of row decoding. It is easily seen that exactly one lost cell per row could be

recovered, but two or more lost cells lead to unrecoverable cell loss. Therefore, the cell loss probability after row decoding (i.e. the residual cell loss probability) is,

$$P_{L,R} = \frac{1}{N} \sum_{i=2}^N i \binom{N}{i} P_l^i (1 - P_l)^{N-i} \quad (2.4)$$

Next, we calculate the post-decoding probability that the cell is correct,  $P_{C,R}$ , and the post-decoding probability that cell is in error,  $P_{E,R}$ . The amount of reduction in the cell loss rate ( $P_L - P_{L,R}$ ) will be added to the other two post-decoding probabilities, with certain percentage as follows. When a lost cell is recovered it will be correct if all the other ( $N-1$ ) cells in the row are correct, whereas if there is at least one cell in error in the remaining ( $N-1$ ) cells in the row, the recovered cell will be in error. The post-decoding probabilities are therefore obtained by modifying the pre-decoding probabilities in (2.1) and (2.2) as follows:

$$P_{C,R} = P_c + \frac{P_c^{N-1}}{\sum_{i=0}^{N-1} P_e^i P_c^{N-i-1}} (P_l - P_{L,R}) \quad (2.5)$$

$$P_{E,R} = P_e + \frac{\sum_{i=1}^{N-1} P_e^i P_c^{N-i-1}}{\sum_{i=0}^{N-1} P_e^i P_c^{N-i-1}} (P_l - P_{L,R}) \quad (2.6)$$



Note that,

$$P_{C,R} + P_{L,R} + P_{E,R} = 1 \quad (2.7)$$

Now we consider the probabilities after the second round of decoding, column decoding, denoted by  $P_{L,RC}$ ,  $P_{C,RC}$  and  $P_{E,RC}$ . It is possible to assume that the statuses of cells in any column are uncorrelated. Hence, by following exactly the same logic of the first round these probabilities can be expressed as:

$$P_{L,CR} = \frac{1}{M} \sum_{i=2}^M i \binom{M}{i} P_{L,R}^i (1 - P_{L,R})^{M-i} \quad (2.8)$$

$$P_{C,CR} = P_{C,R} + \frac{P_{C,R}^{M-1}}{\sum_{i=0}^{M-1} P_{E,R}^i P_{C,R}^{M-i-1}} (P_{L,R} - P_{L,CR}) \quad (2.9)$$

$$P_{E,CR} = P_{E,R} + \frac{\sum_{i=1}^{M-1} P_{E,R}^i P_{C,R}^{M-i-1}}{\sum_{i=0}^{M-1} P_{E,R}^i P_{C,R}^{M-i-1}} (P_{L,R} - P_{L,CR}) \quad (2.10)$$

The subscript R and CR denotes to row and row column decoding respectively.

### **2.2.3 Discussion and Results:**

The post-decoding cell loss rate based on the above analysis is given in Figure 2.5 for different matrix sizes. It is clear that a smaller matrix performs better than a larger matrix. This is natural because of the increased redundancy in a smaller matrix. For example the redundancy in an  $8 \times 8$  matrix is 23 % while it is 12 % in  $16 \times 16$  matrix and 6 % in a  $32 \times 32$  matrix. Figure 2.5 shows the improvement of going from single round of decoding, i.e. row wise decoding, to two rounds of decoding, i.e. row-column-wise decoding. The matrix size of  $16 \times 16$  will be adopted for the rest of the thesis unless mentioned otherwise.

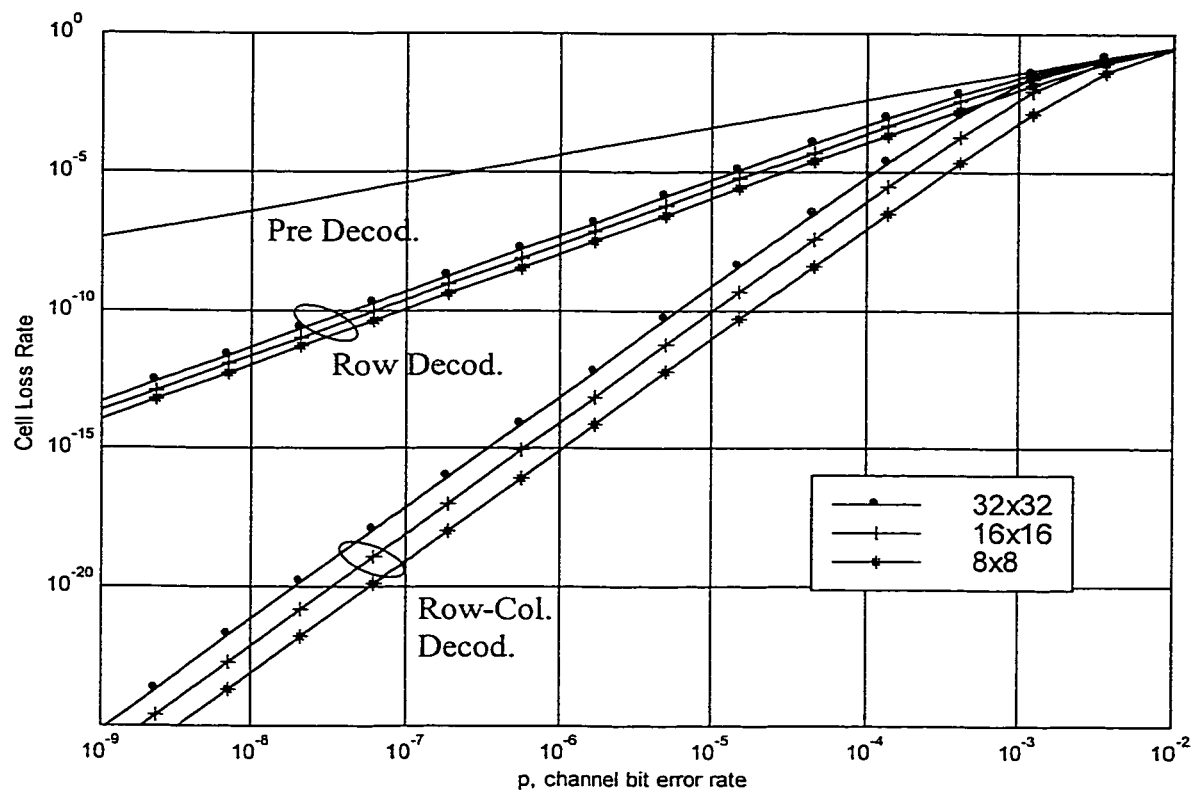


Figure 2.5: Studying the effect of different matrix size on post decoding Cell loss rate

For a matrix size  $16 \times 16$  Figure 2.5 shows the improvement in recovering cells from non-coded case to decoding row-wise and row-column-wise. Thus, Figure 2.5 provides a good indication of the powerfulness of this simple coding scheme in recovering lost cells. The post-decoding CLR is many orders of magnitude smaller than the pre-decoding CLR.

The pre-decoding and post-decoding cell error rate (after two rounds of decoding) is plotted in Figure 2.6. It is observed that the post-decoding error rate is very close to the pre-decoding error rate, thus giving an indication that most of the recovered cells are recovered correctly. At high bit error rate it appears in the figure that the post decoding cell error rate is higher than that of pre-decoding showing that some of the recovered cells are recovered in error.

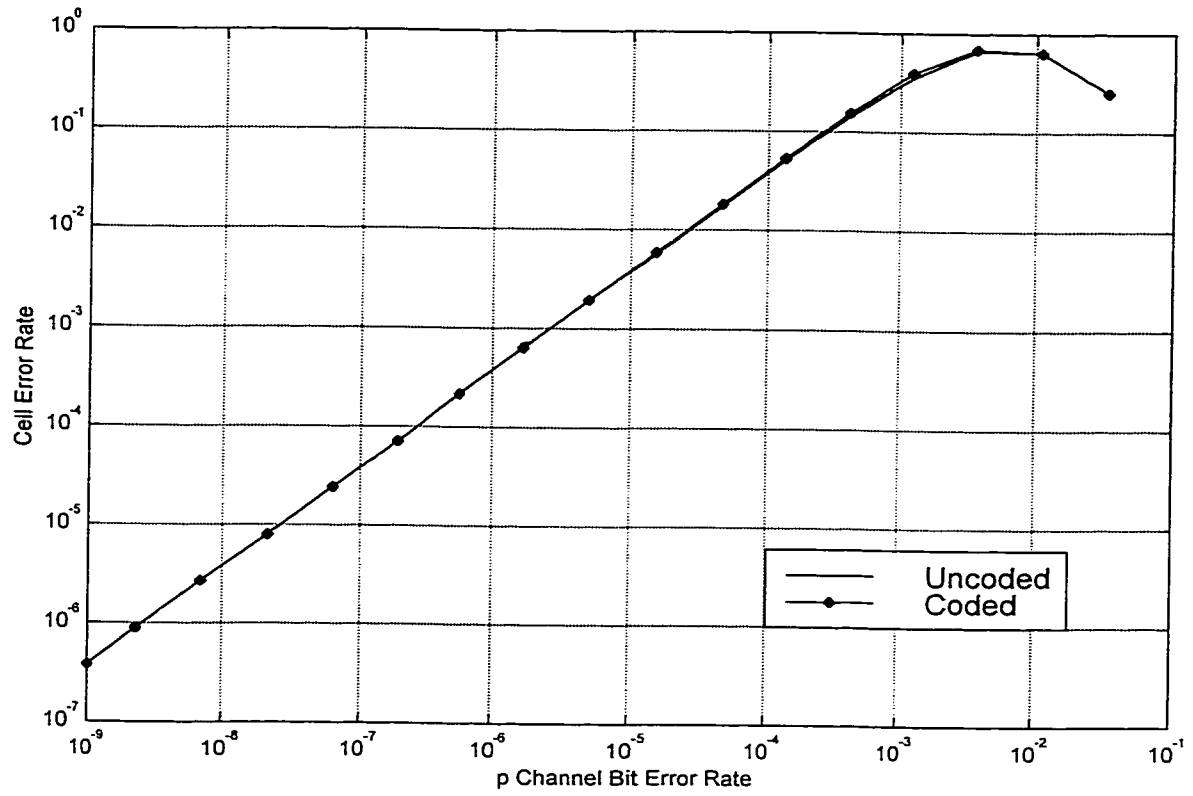


Figure 2.6: Cell error rate performance for 16x16 matrix code

For matrix size  $16 \times 16$ , the cell loss rate performance of the code after row-wise and column-wise decoding is compared to the performance using Bound-III (B-III) in [34] is given in Figure 2.7. The work in [34] introduces a new concept of basic patterns, from which all cell loss patterns may be generated. The pattern is the possible shape or arrangement of lost cells in a decoding matrix. For a given number of lost cells these basic patterns are classified to recoverable basic patterns and unrecoverable basic patterns. And using the unrecoverable basic patterns three bounds was derived for the post-decoding cell loss rate, each with some degree of tightness. The B-III obtained in [34] is giving the theoretical limit on the capability of the code. The comparison is performed to validate the analytical results obtained in this work. The figure shows difference in B-III and the results obtained in this work, and this is because of the following reasons:

- The bound obtained in [34] is obtained by performing row and column-wise decoding in many rounds until all cells are recovered or no further recovery is possible, thus it gives the optimum decoding capability. While in our study, only two rounds of decoding are performed, i.e. row-wise and column-wise.
- The assumption of uncorrelation between the two rounds of decoding.

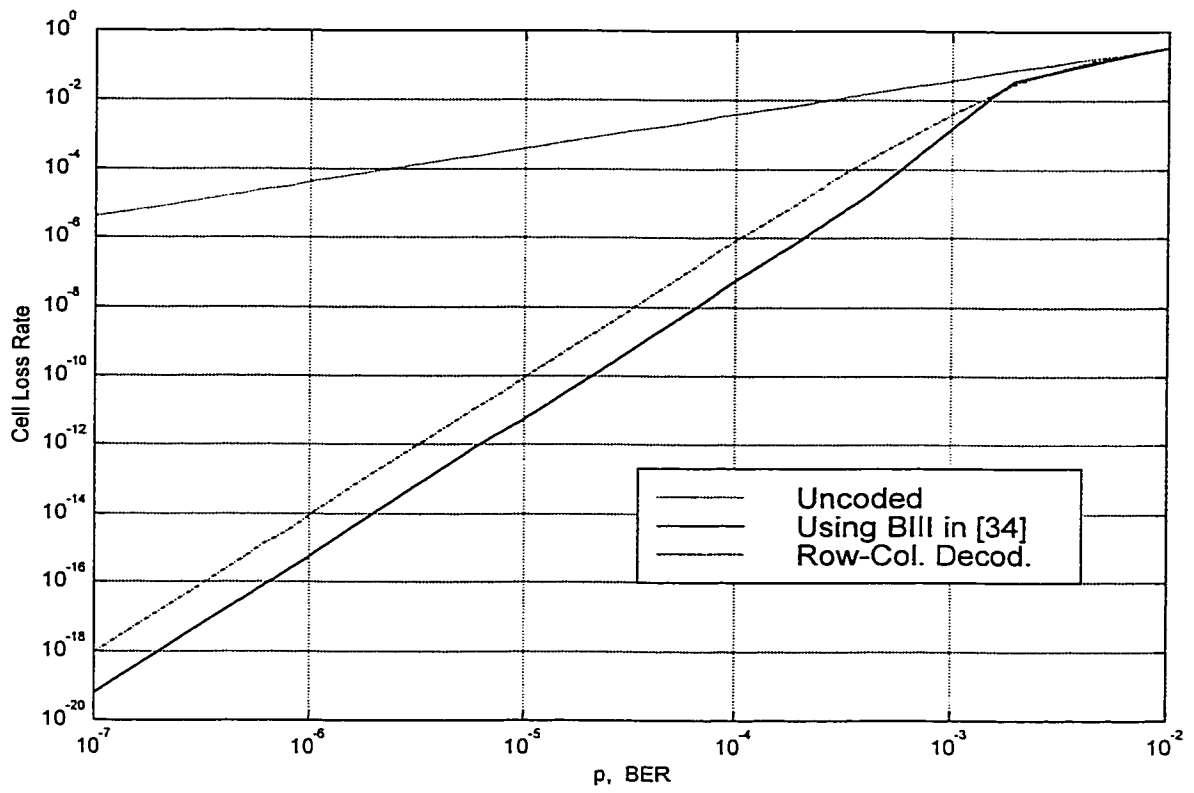


Figure 2.7: Comparison of row-column-wise decoding and BIII in [34]

### 2.3 Correlated Channel Case:

The effect of correlated channel is studied in this section, where noise samples are assumed to be correlated. The correlated channel is modeled using first order two-state Markov process specified by the transition probabilities, as shown in Figure 2.8. In this model the channel is assumed to be either in the “correct” state where the transmission is error-free, or in the “error” state where the transmission is always erroneous. The steady state behavior, described by the limiting probabilities of the “1” and “0” states, is given by [36]:

$$x_0 = \Pr[\text{correct}] = \frac{1 - x_{11}}{2 - x_{00} - x_{11}} \quad (2.11)$$

$$x_1 = \Pr[\text{error}] = \frac{1 - x_{00}}{2 - x_{00} - x_{11}} \quad (2.12)$$

Thus the channel bit error rate  $p$  is given by  $x_1$ . The transition probabilities of the channel model are obtained for given correlation coefficient and channel bit error rate.

The correlation coefficient  $\omega$  of the process is given by [35,36],

$$\omega = |x_{00} + x_{11} - 1| \in (0,1) \quad (2.13)$$

where  $x_{ij}$  is the transition probability from state  $i$  to state  $j$ ,  $i, j=0,1$ .



In general, the closer  $\omega$  to 1 is, the burstier the channel becomes. For example, when  $p \in (0, 0.1]$ , equation (2.12) and (2.13) imply  $x_{00} \in [0.92, 1)$  and  $x_{11} \in [0.2, 0.28)$  for  $\omega = 0.2$ , meaning the errors tend to be dispersed over time. On the other hand, we have  $x_{00} \in [0.99, 1)$  and  $x_{11} \in [0.9, 0.91)$  for  $\omega = 0.9$ , and errors are likelier to be in groups, i.e. correlated. Figure 2.9, shows a typical realization of the channel error pattern obtained over 200-slot periods for  $p = 0.05$  and  $p = 0.1$ , illustrating the respective random and bursty characteristics for  $\omega = 0.2$  and  $\omega = 0.9$ .

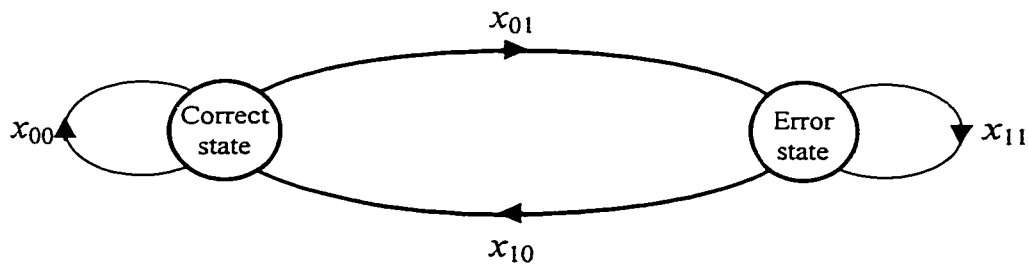


Figure 2.8: Two state Markov process correlated channel model

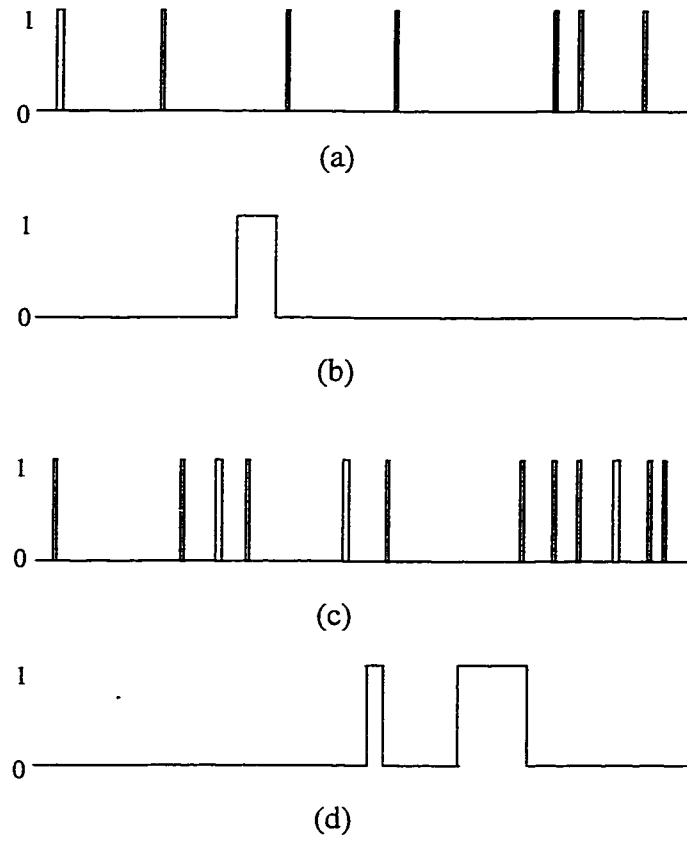


Figure 2.9: Channel error pattern over 200-slot periods for the following channel bit error rate and correlation coefficient: (a)  $p = 0.05, \omega = 0.2$ ; (b)  $p = 0.05, \omega = 0.9$ ; (c)  $p = 0.1, \omega = 0.2$ ; (d)  $p = 0.1, \omega = 0.9$

### **2.3.1 Simulation of Correlated Channel:**

There is no closed analytical form for the probabilities with correlation. Therefore, simulation is used to determine the pre-decoding probabilities with correlation effect. The evaluated pre-decoding probabilities from the simulation are applied to Equations (2.4-2.10) to evaluate the two-dimensional matrix code performance with correlated channel. The flow chart used for correlated channel simulation to obtain the pre-decoding probabilities is shown in Figure 2.10.

In the initialization state we specify the channel bit error rate and the correlation coefficient  $\omega$ . The correlation coefficient is taken to be  $\omega=0.75$  throughout the thesis. Taking the values of the channel bit error rate and the correlation coefficient the transition probabilities are evaluated using Equation 2.12 and Equation 2.13. The bits are generated which represents correct or error bit and filled in a cell that has the same size as ATM cell. When the channel cell is filled the header and payload portions are checked. Four different scenarios are possible: header is correct and payload is correct (HCPC), header is correct and payload is erroneous (HCPE), header is erroneous and payload is correct (HEPC) and header is erroneous and payload is erroneous (HEPE). The ATM cell will be lost if the header is in error, i.e. the cases (HEPC and HEPE), thus this will give the pre-decoding cell loss rate. The received cell will be in error if the payload portion is in error and the header is correct, i.e. (HCPE) case, thus this will give the cell error rate as follows,

$$P_l = \frac{n_{HEPC} + n_{HEPE}}{T} \quad (2.14)$$

$$P_e = \frac{n_{HCPE}}{T} \quad (2.15)$$

$$P_c = 1 - P_e - P_l \quad (2.16)$$

where,

$T$  = total number of cells in simulation

$n_{HEPC}$  = number of cells where header is in error and payload is correct

$n_{HEPE}$  = number of cells where header is in error and payload is in error

$n_{HCPE}$  = number of cells where header is in error and payload is correct

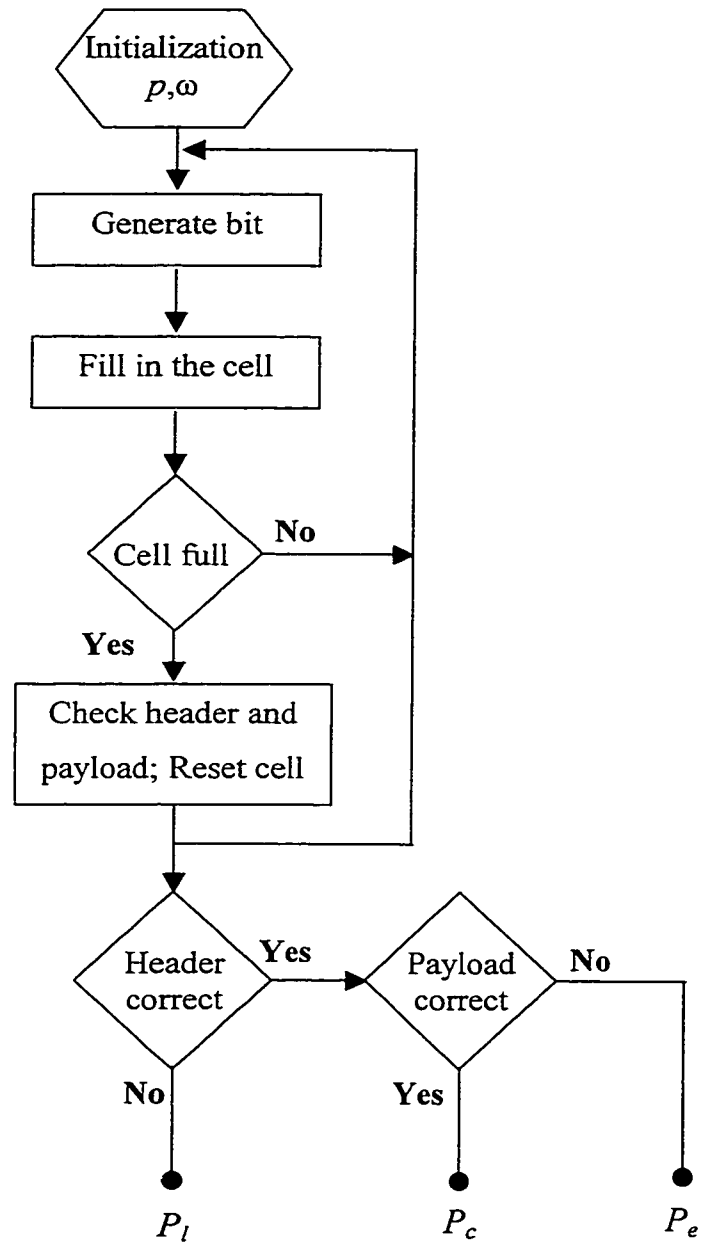


Figure 2.10: Flow chart of correlated channel

### **2.3.2 Discussion and Results:**

The results of the simulation for the pre-decoding cell loss and cell error rate is given in Figure 2.11 and 2.12 respectively. For the sake of comparison, the results of pre-coding probabilities for random channel are also shown in the same figures. The figures show that both cell loss rate and cell error rate are smaller for the correlated channel as compared to the random channel at the same bit error rate. This can be explained as follows. The correlated channel gives some kind of relation in the errors in the payload and header, such that when header is affected by channel errors the payload is also affected by a probability that is higher than that of the random channel case. Therefore, when a cell is lost it is most likely to be in error. On the other hand, the percentage of surviving cells that are correct is higher. Also, for the correlated channel, since errors occur in bursts less cells will be effected by errors for the same channel bit error rate, which gives better pre-decoding cell loss and error rates.

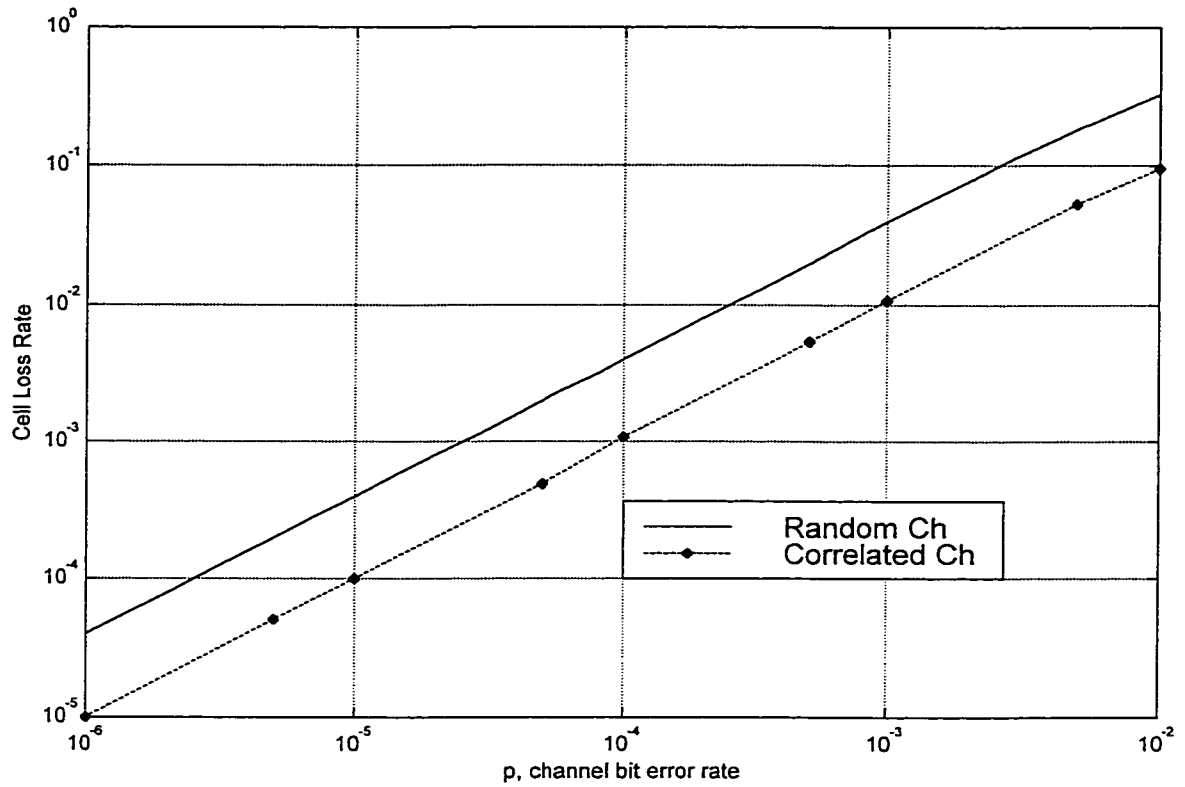


Figure 2.11: Comparison of random channel and correlated channel for pre-decoding cell loss rate ( $\omega=0.75$ )

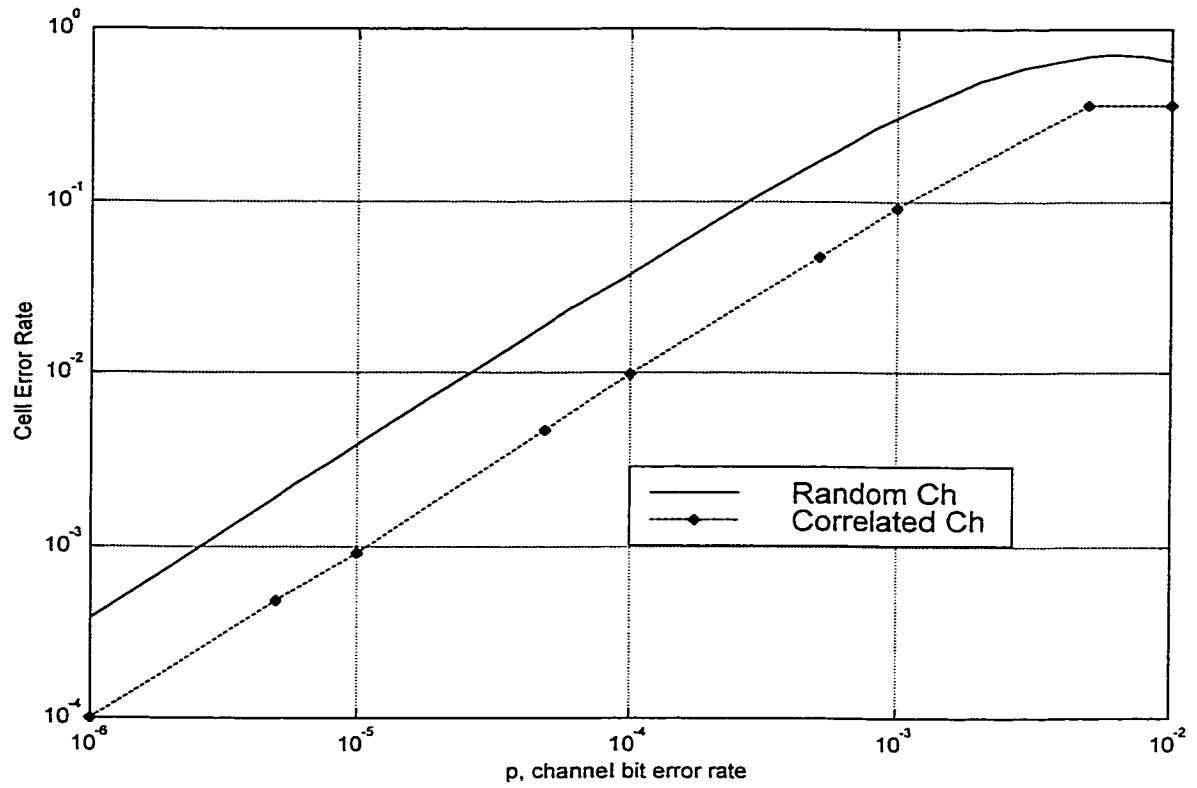


Figure 2.12: Comparison of random channel and correlated channel for pre-decoding cell error rate ( $\omega=0.75$ )



To verify the results of the simulation, the correlation coefficient was set to zero, i.e. uncorrelated channel. The results are presented in Figure 2.13 and Figure 2.14 for cell loss and cell error rate respectively, and compared with the case of random channel. The comparison shows identical results for both simulated and analytical random channel result for pre-decoding cell loss and error rates.

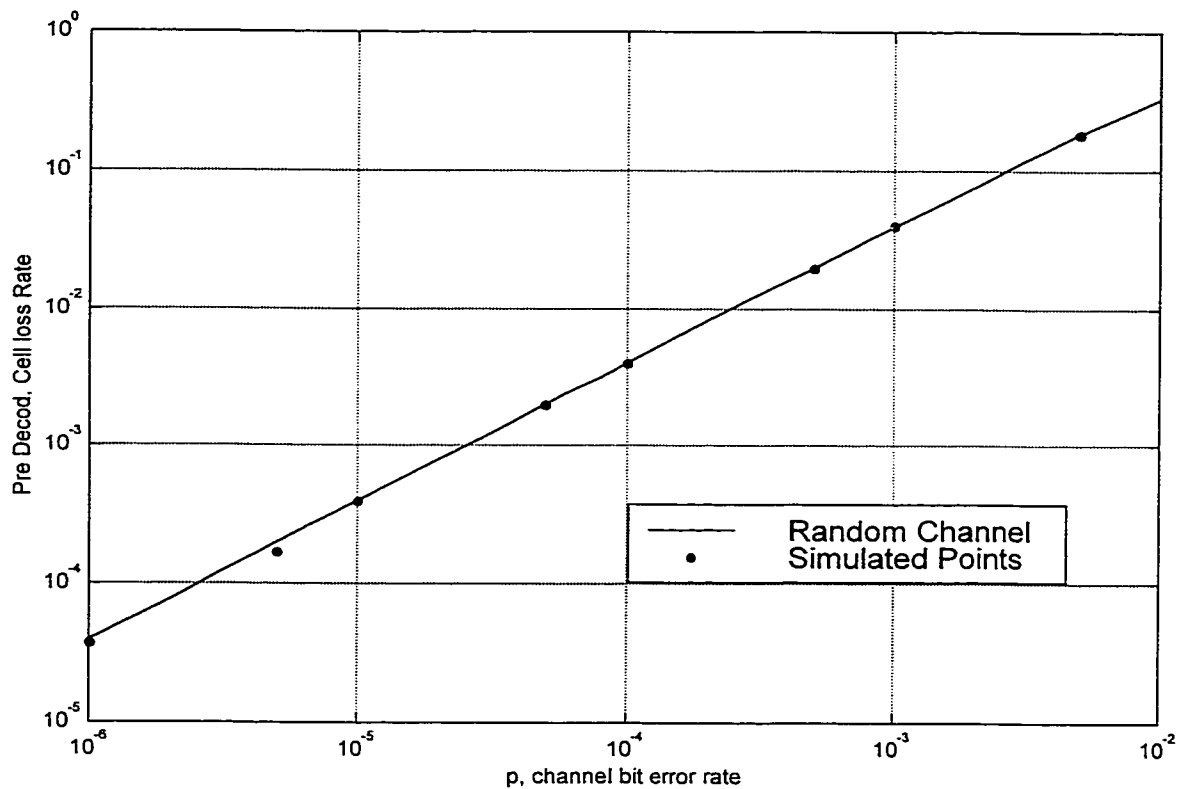


Figure 2.13: Comparison between random channel and simulated correlated channel with correlation coefficient  $\omega = 0$ , for pre-decoding cell loss rate

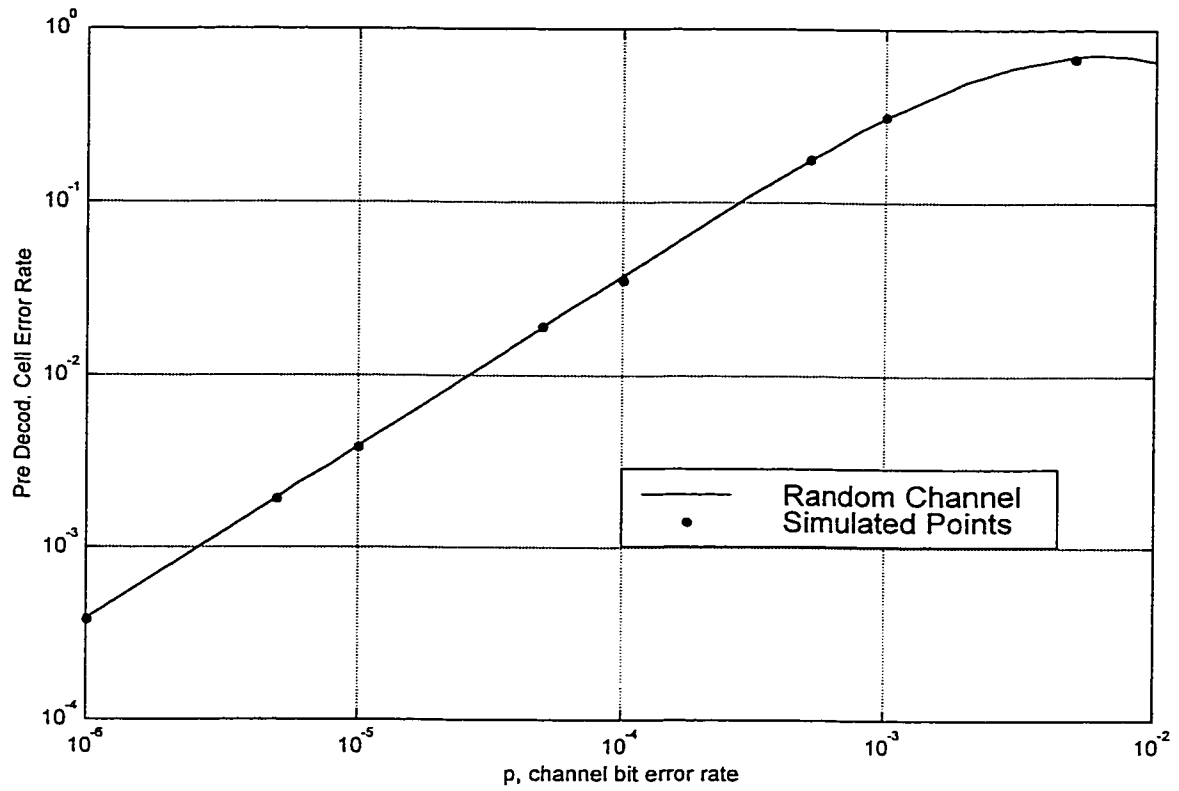


Figure 2.14: Comparison between random channel and simulated correlated channel with correlation coefficient  $\omega = 0$ , for pre-decoding cell error rate

The pre-decoding values obtained using the simulation are substituted in Equations (2.4-2.10) to obtain the post-decoding cell loss and error rate probabilities. Note that Equation 2.5 and others assume random cell loss. This model is valid even at correlated channel case because cells are transmitted in a direction opposite to that at which they will be first decoded. Also the length of the cell is large (424 bits), therefore it is possible to assume that the errors are independent from cell to cell. The results of post-decoding probabilities are shown in Figure 2.15 and Figure 2.16 for correlated channel. For comparison the performance of random channels are shown in the same figures. As expected the Figure 2.15 shows that a large percentage of the lost cells were recovered. The post-decoding cell error rate is very close to the pre-decoding one as shown in Figure 2.16, which shows that most of the recovered cells are correct. Also the figures shows that the performance of correlated channel is better than that of random channel for the same bit error rate.

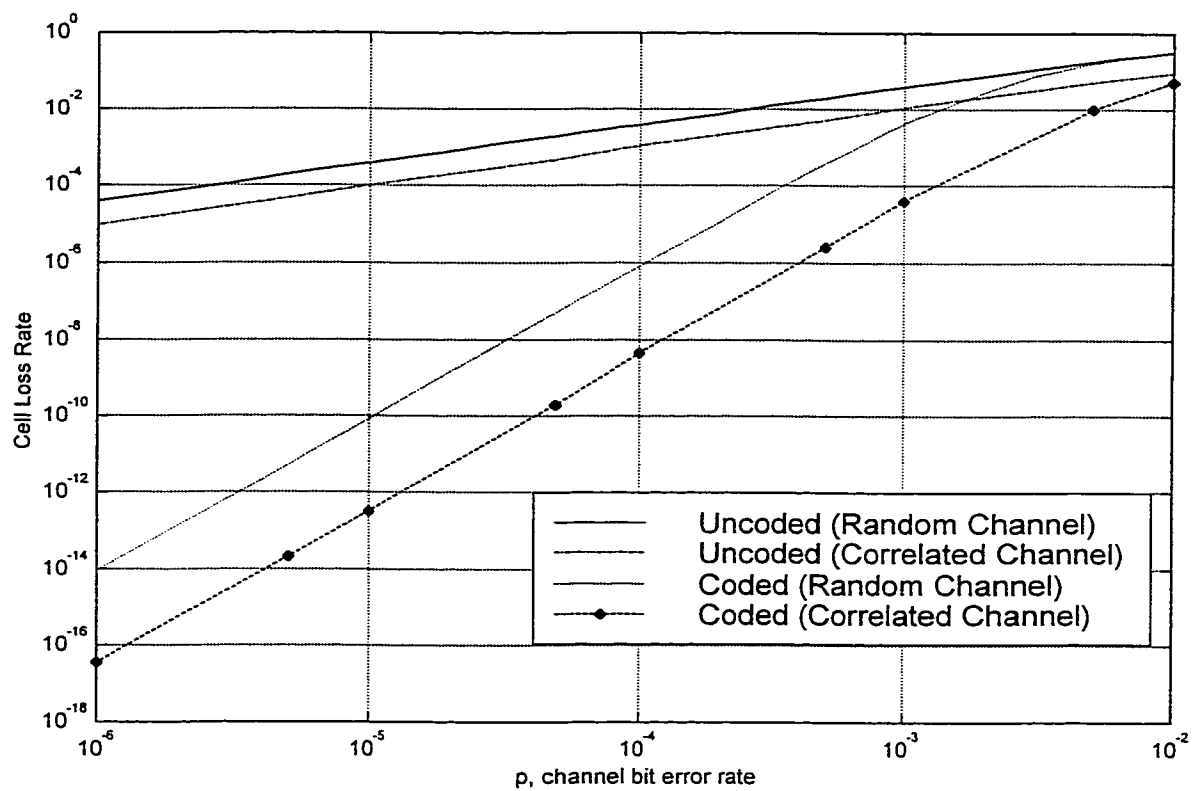


Figure 2.15: Cell loss performance for random channel and correlated channel with correlation coefficient  $\omega = 0.75$

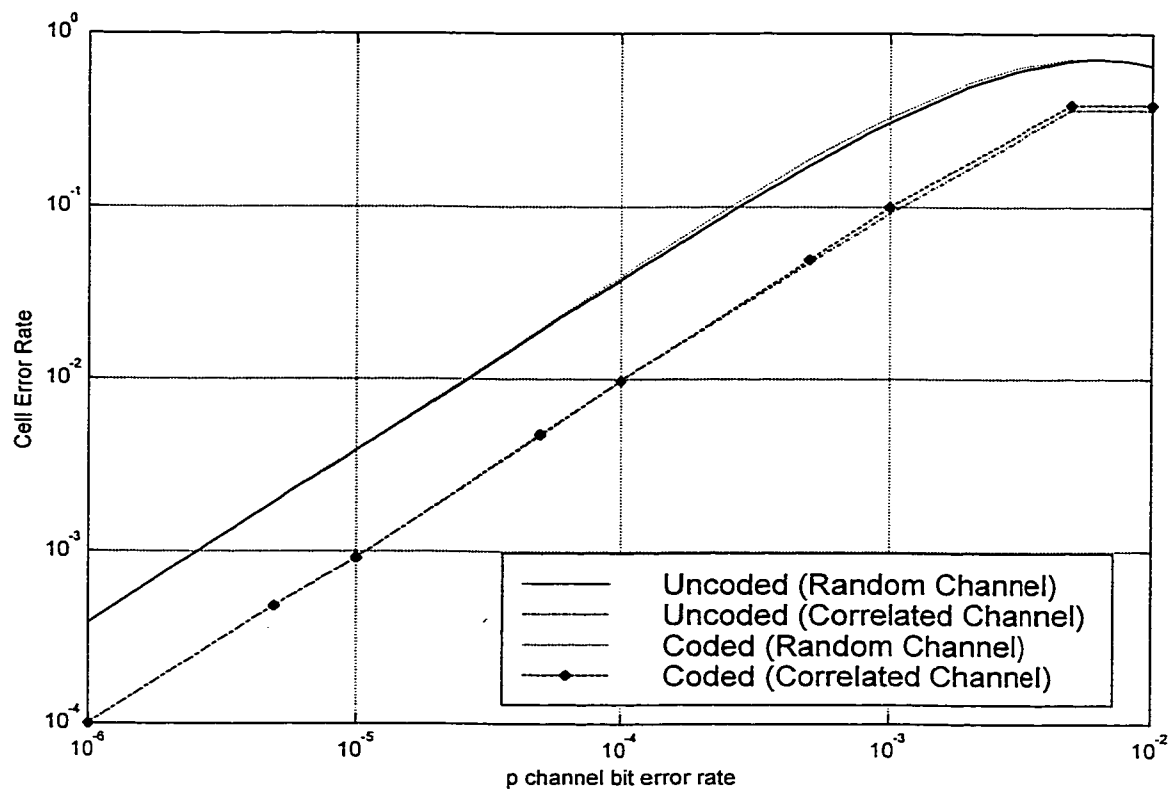


Figure 2.16: Cell Error performance for random channel and correlated channel with correlation coefficient  $\omega = 0.75$

## 2.4 Simulation Results:

Simulation of the two-dimensional matrix code has been performed to confirm the results of the analysis. The simulation result of the random channel cell loss and cell error rate is shown in Figure 2.17 and Figure 2.18 respectively, which shows the pre-decoding and post-decoding probabilities. The effect of channel burstiness is shown in Figure 2.19 and Figure 2.20, for cell loss and error rates respectively. The figures show that the simulated results agrees with the analytical results. Therefore, the analysis gives a true measure of the code performance.

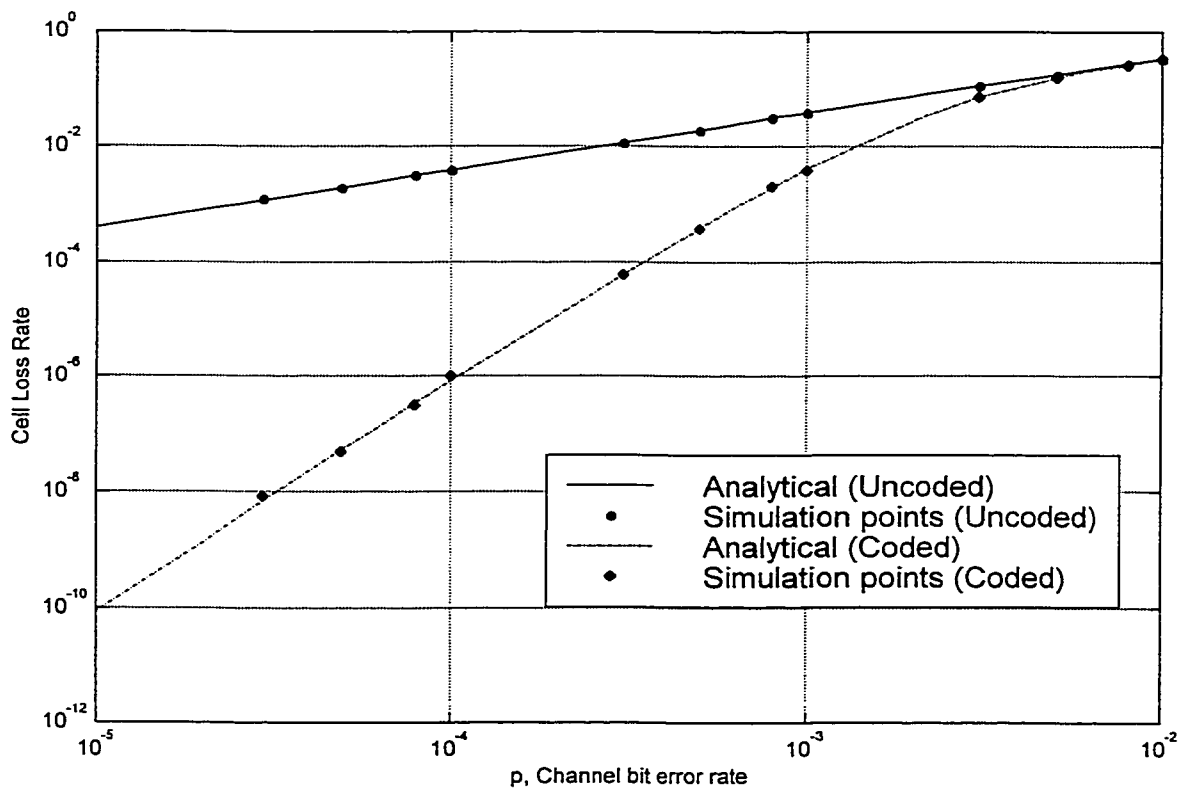


Figure 2.17: Simulation and analytical results of cell loss rate for random channel

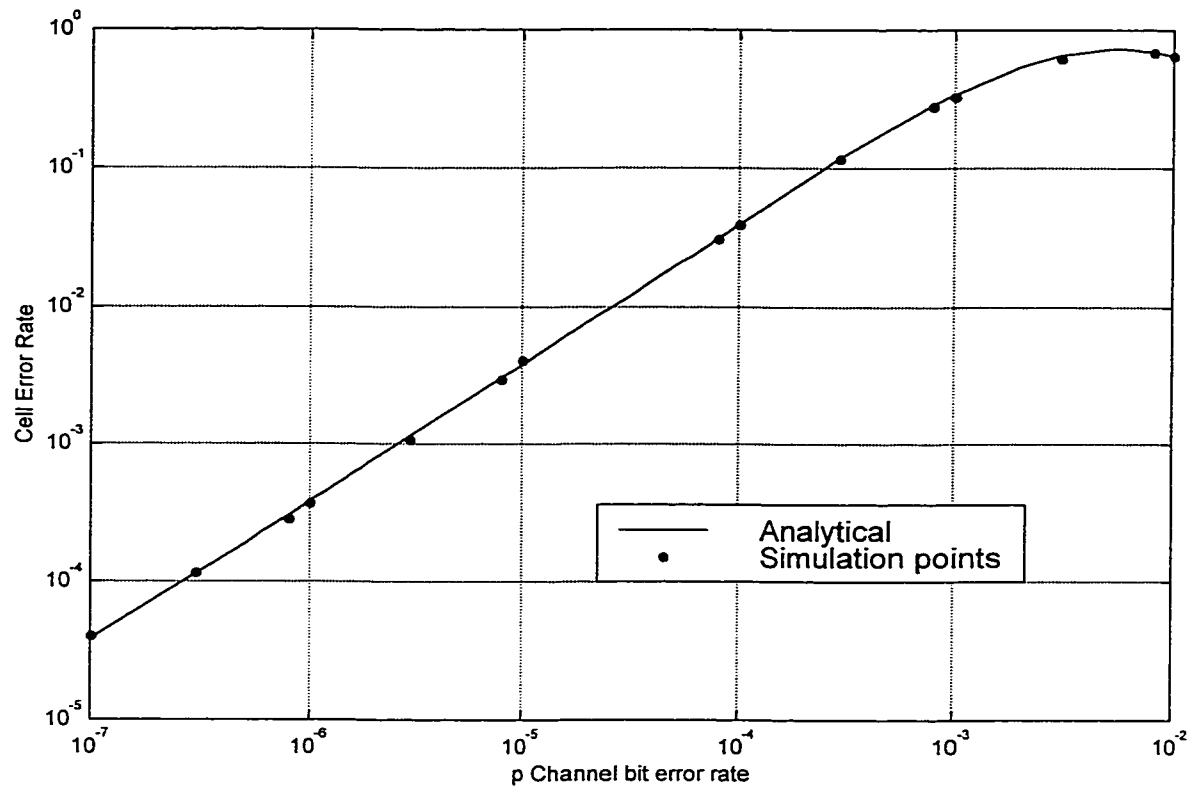


Figure 2.18: Simulation and analytical results of Coded cell error rate for random channel

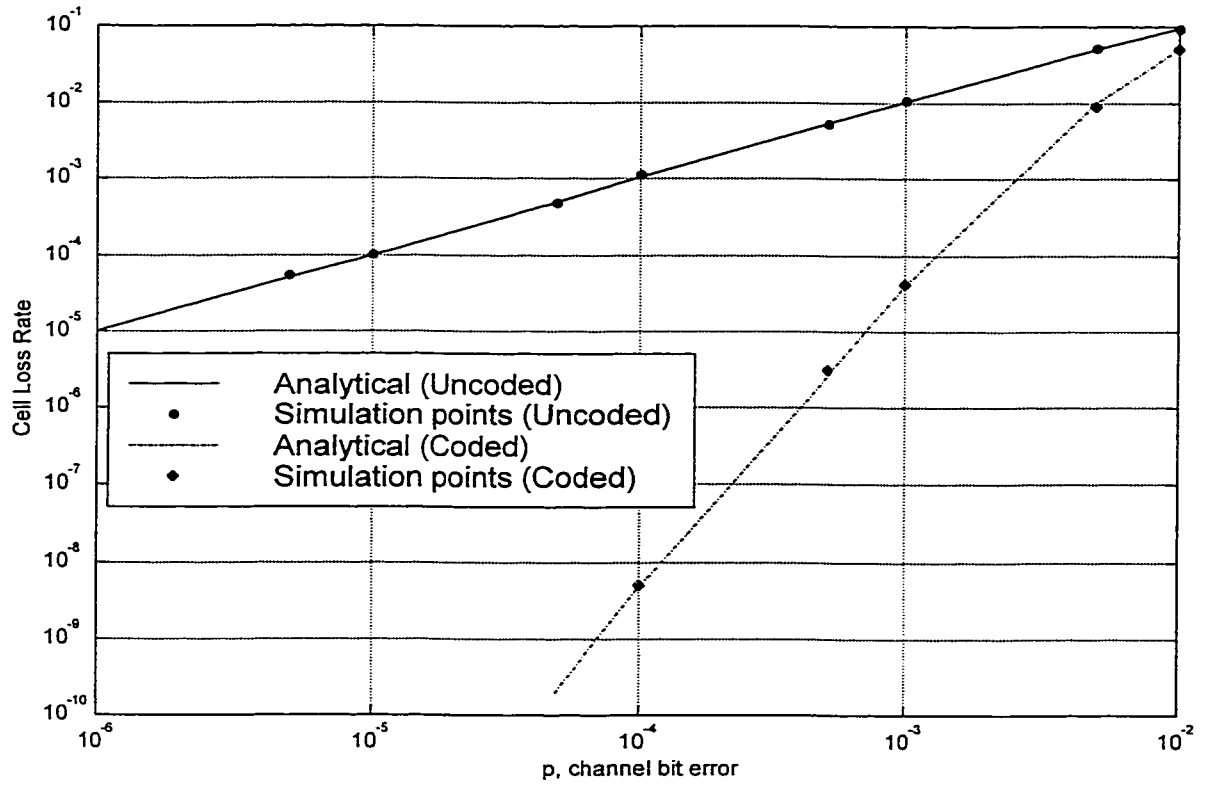


Figure 2.19: Simulation and analytical results cell loss rate for correlated channel with  $\omega=0.75$



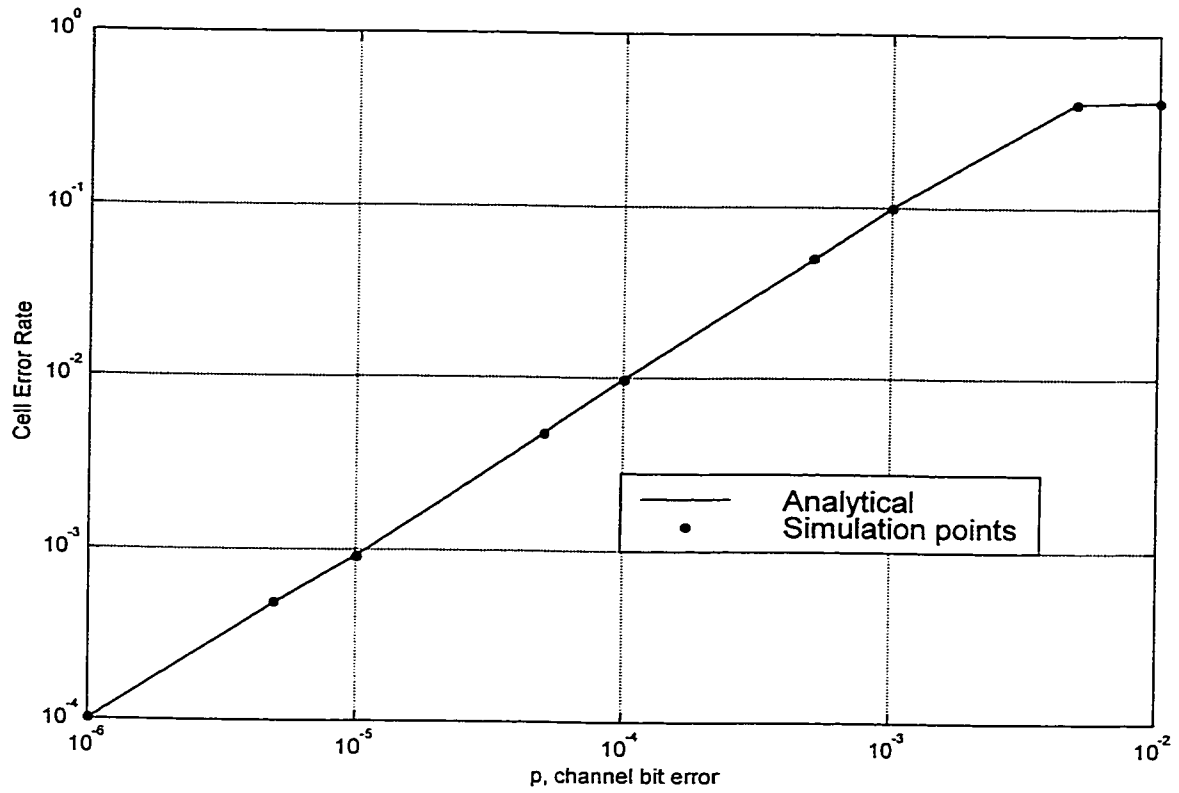


Figure 2.20: Simulation and analytical results of Coded cell error rate for correlated channel with  $\omega=0.75$

## **2.5 Summary:**

In this chapter, we presented a detailed analysis of the code performance over noisy ATM link in terms of cell loss and cell error rate. The cell loss due to channel errors is considered in this chapter by assuming infinite buffer size. The effect of both random channel errors and correlated channel errors were investigated and compared. The pre-decoding probabilities of correlated channel are found using simulation, because there is no analytical formula that gives the effect of correlated channel on ATM cell, and these pre-decoding probabilities are used to evaluate the code performance. Finally, simulation was performed to verify the analytical results. In the following chapter, we will include cell loss due to buffer overflow and study the effect of finite buffer size on the code performance.

# CHAPTER 3

## RELIABILITY STUDY: FINITE BUFFER CASE

### 3.1 Introduction

One of the major sources of errors in the ATM systems is the buffer overflow, which results in ATM cell losses that degrade the quality of service. In this chapter the cell losses due to buffer overflow, in addition to the losses due to channel errors, will be handled. The performance of the two-dimension matrix code will be analyzed with the effect of finite buffer on the system for random traffic loading, and simulation is performed for the case of bursty traffic loading.

Consider the case where the FEC two-dimensional matrix code is applied to a certain source traffic when it traverses a poor link. Denote this traffic by tagged traffic. The traversing cells are arrived at the input of an  $X \times Y$  ATM switch, their header is checked for errors, and a cell with erroneous header is discarded. The cells that survive channel errors are forwarded to the appropriate output. Output buffering scheme is adopted in the structure of the ATM switch fabric, where all the queuing is done at the outputs of the switch with a separate FIFO buffer provided for each output as shown in Figure 3.1. At the output port cells are buffered for further processing, and the cells are discarded when the buffer is full. The tagged cells that survive both channel errors and buffer overflow are filled in the matrix and the missing cells are replaced by the all-zero dummy cell. Decoding is performed to recover lost cells as explained in Chapter 2.

The performance of the two-dimension matrix code is studied in terms of cell loss and cell error rates. In the process we investigate the interactive effects of several parameters such as traffic intensity, buffer size, burstiness of the traffic, and channel bit error rate on the performance of the code.

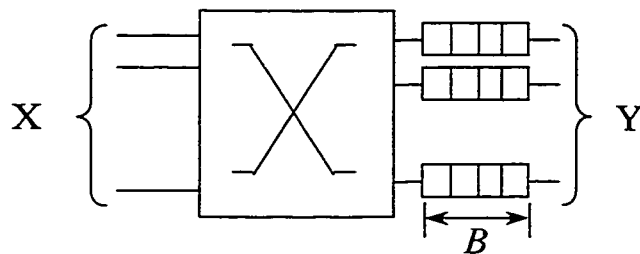


Figure 3.1: Output Buffered  $X \times Y$  ATM switch

## **3.2 Random Traffic Loading**

In the following, we evaluate and investigate the performance of the FEC matrix code for random traffic loading with a buffer of finite size in addition to the effect of channel errors.

### **3.2.1 Pre-decoding probabilities:**

The analysis is similar to that carried in the infinite buffer case in the previous chapter. The only difference will be in the pre-decoding probabilities, which will be modified to reflect the effect of buffer overflow on the performance. It is assumed that the cell discard due to buffer overflow and that due to channel errors are independent.

Let  $f$  denote the probability of cell loss due to buffer overflow.

A cell will be received correctly with probability  $P_{c,f}$  ( $f$  stands for finite buffer ) if it survives both the physical layer errors and buffer loss, and is given as

$$P_{c,f} = P_c(1 - f) \quad (3.1)$$

where  $P_c$  is the probability that the cell survives channel errors, which was calculated in Chapter2.

Assuming all errors in the header portion are detectable, a received cell will be in error when the cell encounters no buffer loss and the payload portion is received in error, thus the probability of error  $P_{e,f}$  is given as,

$$P_{e,f} = (1 - f)P_e \quad (3.2)$$

where  $P_e$  is the probability that the payload portion of the survived cell is affected by channel errors.

Using the pre-decoding cell error and correct probabilities; the probability that the cell is lost is then given by

$$P_{l,f} = 1 - P_{c,f} - P_{e,f} \quad (3.3)$$

To evaluate the buffer overflow probability we adopted the random traffic loading model in [37]. The cell arrivals on the  $X$  switch inputs, for  $X \times Y$  ATM switch, are modeled by independent and identical Bernoulli processes. With output buffering, all queuing is done at the outputs with a separate FIFO queue of size  $B$  cells at each output port of the ATM switch fabric. In the analysis, we fix our attention on a particular (i.e., tagged) output queue. Defining the random variable  $A$  as the number of cell arrivals destined for the tagged output in a given time slot, then the probability,  $a_k$ , of  $k$  cells arriving at given

time slot, which is the unit time required to transmit a cell over the output link, at the tagged output is given by,

$$a_k \stackrel{\Delta}{=} \Pr[A = k] = \binom{X}{k} (\rho/X)^k (1 - \rho/X)^{X-k}$$

$$k = 0, 1, \dots, X \quad (3.4)$$

where  $\rho$  is the total normalized traffic load input to the switch. It is assumed that the effect of the channel on the traffic intensity that reaches the buffer is negligible. That is, those cells that are discarded because their headers are wrong do not modify the traffic of the tagged sources.

Let  $Q_m$  denote the number of cells in the tagged queue at the end of the  $m$ th time slot. It is then given by

$$Q_m = \min\{\max(0, Q_{m-1} + A_m - 1), B\} \quad (3.5)$$

where  $A_m$  denotes the number of packet arrivals during the  $m$ th time slot.

When  $Q_{m-1} = 0$  and  $A_m > 0$ , one of the arriving cells is immediately transmitted during the  $m$ th time slot; i.e., a cell flows through the switch without suffering any delay. For finite  $X$  and  $B$ ,  $Q_m$  is modelled by a finite-state, discrete-time Markov chain with state transition probabilities  $P_{ij} \stackrel{\Delta}{=} \Pr[Q_m = j | Q_{m-1} = i]$  given by [37],

$$P_{ij} = \begin{cases} a_0 + a_1 & i = 0, j = 0 \\ a_0 & 1 \leq i \leq B, j = i - 1 \\ a_{j-i+1} & 1 \leq j \leq B-1, 0 \leq i \leq j \\ \sum_{m=j-i+1}^j a_m & j = B, 0 \leq i \leq j \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

where  $a_k$  is given by (3.4). The steady-state queue size can be obtained directly from the Markov chain balance equations to yield

$$q_1 \stackrel{\Delta}{=} \Pr[Q = 1] = \frac{(1 - a_0 - a_1)}{a_0} \cdot q_0 \quad (3.7)$$

$$q_n \stackrel{\Delta}{=} \Pr[Q = n] = \frac{(1 - a_1)}{a_0} \cdot q_{n-1} - \sum_{k=2}^n \frac{a_k}{a_0} \cdot q_{n-k} \quad 2 \leq n \leq B \quad (3.8)$$

where

$$q_0 \stackrel{\Delta}{=} \Pr[Q = 0] = \frac{1}{1 + \sum_{n=1}^B q_n / q_0} \quad (3.9)$$

A cell will not be transmitted on the tagged output line during the  $m$ th time slot if, and only if,  $Q_{m-1} = 0$  and  $A_m = 0$ . Therefore, letting  $U_0$  denote the normalized switch throughput, we have [37]



$$U_0 = 1 - q_0 a_0 \quad (3.10)$$

A cell will be lost if it finds the output buffer already containing  $B$  cells. Dividing the utilization of the output line  $U_0$  by the total offered load  $\rho$ , we obtain the cell success probability. Therefore, the buffer overflow probability is given by [37],

$$f = 1 - \frac{U_0}{\rho} \quad (3.11)$$

### **3.2.2 Post Decoding Probabilities:**

The post decoding probabilities for  $M \times N$  matrix size are the same as those for the infinite buffer case Equations (2.4-2.10), and they are rewritten here for convenience

$$P_{L,CR} = \frac{1}{M} \sum_{i=2}^M i \binom{M}{i} P_{L,R}^i (1 - P_{L,R})^{M-i} \quad (3.12)$$

$$P_{C,CR} = P_{C,R} + \frac{P_{C,R}^{M-1}}{\sum_{i=0}^{N-1} P_{E,R}^i P_{C,R}^{M-i-1}} (P_{L,R} - P_{LC,R}) \quad (3.13)$$

$$P_{E,CR} = P_{E,R} + \frac{\sum_{i=1}^{M-1} P_{E,R}^i P_{C,R}^{M-i-1}}{\sum_{i=0}^{M-1} P_{E,R}^i P_{C,R}^{M-i-1}} (P_{L,R} - P_{L,CR}) \quad (3.14)$$

where  $P_{L,R}$ ,  $P_{C,R}$  and  $P_{E,R}$  are given by,

$$P_{L,R} = \frac{1}{N} \sum_{i=2}^N i \binom{N}{i} P_{l,f}^i (1 - P_{l,f})^{N-i} \quad (3.15)$$

$$P_{C,R} = P_{c,f} + \frac{P_{c,f}^{N-1}}{\sum_{i=0}^{N-1} P_{e,f}^i P_{c,f}^{N-i-1}} (P_{l,f} - P_{L,R}) \quad (3.16)$$

$$P_{E,R} = P_{e,f} + \frac{\sum_{i=1}^{N-1} P_{e,f}^i P_{c,f}^{N-i-1}}{\sum_{i=0}^{N-1} P_{e,f}^i P_{c,f}^{N-i-1}} (P_{l,f} - P_{L,R}) \quad (3.17)$$

### **3.2.3 Discussion and Results:**

The performance of the code will be investigated for  $16 \times 16$  matrix size and  $X=16$  inputs to the ATM switch through out this chapter unless stated otherwise. The cell loss performance for the code is shown in Figure 3.2 for random channel with buffer size of  $B=16$  cells and random traffic loading with normalized traffic intensity  $\rho = 0.8$ . The cell loss rate for uncoded and coded systems are shown in the figure. The high capability of the code in recovering the lost cells is obvious. The figure also shows the curves for the infinite buffer case. It can be seen that the curves of finite buffer case exhibit a floor effect, which occurs because of buffer overflow.

At the floor region (low bit error rate) of the curves the cell loss probability due to buffer overflow dominates the cell loss probability due to channel errors hence no effect is observed for variation in channel bit error rate. On the other hand, at higher bit error rates, the cell loss due to channel errors becomes more dominant and therefore the finite buffer case exhibits essentially the same performance as the infinite buffer case.

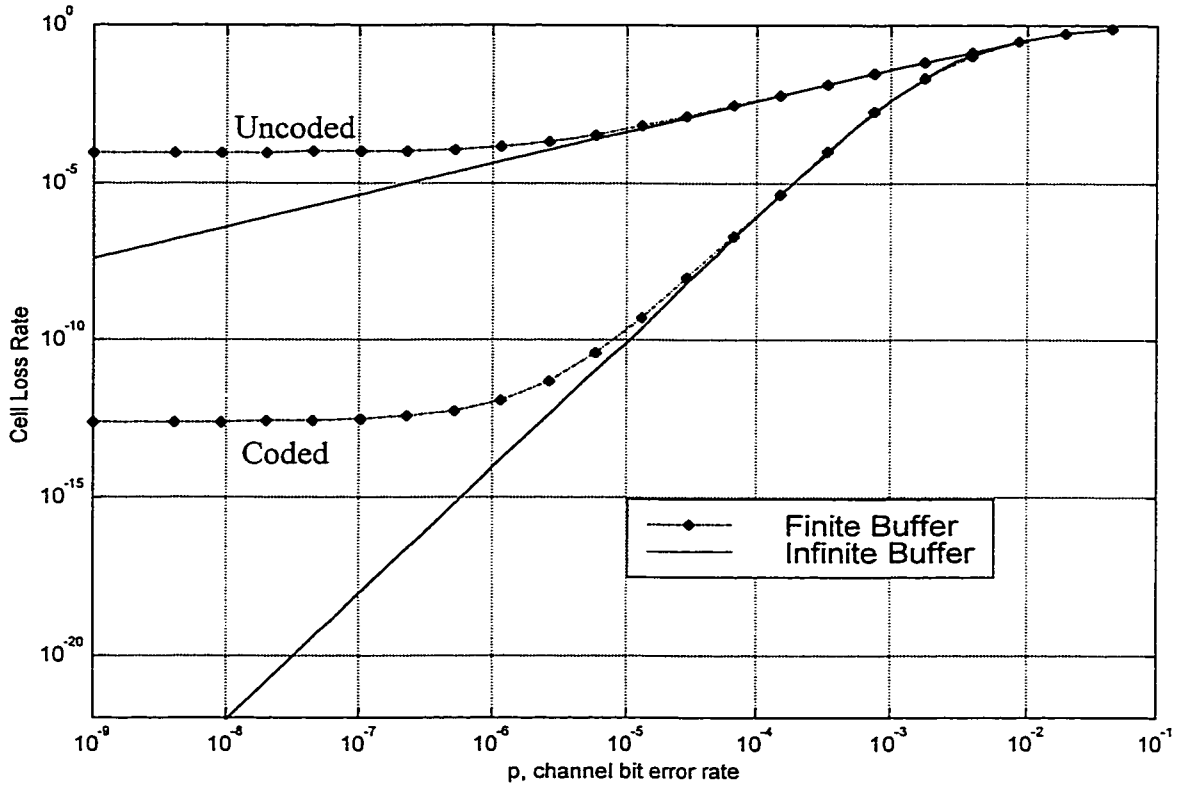


Figure 3.2 Comparison of Cell loss rate for random channel between infinite buffer case and finite buffer with ( $B=16$ ) at  $\rho=0.8$

The cell error rate performance comparison between infinite and finite buffer sizes is shown in Figure 3.3 for the coded system. The figure shows that the buffer has no effect on the cell error performance. This is quite expected because the buffer causes cells to be lost but does not affect the data within the cell.

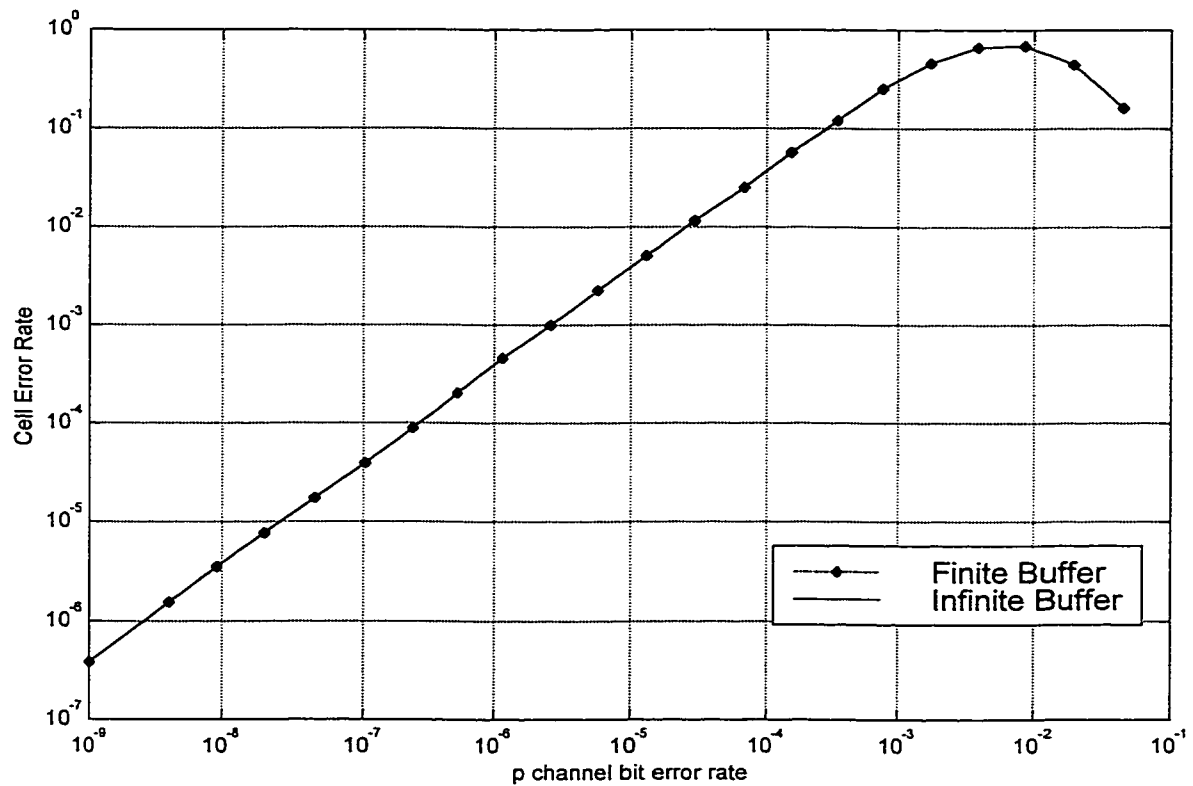


Figure 3.3 Comparison of Cell error performance for random channel between infinite buffer case and finite buffer ( $B=16$ ) at  $p=0.8$

The effect of increasing the buffer size from  $B=16$  to  $B=32$  is shown in Figure 3.4 for  $\rho = 0.8$ . The figure shows that by doubling the buffer size the performance has been greatly improved. In fact the performance approaches that of infinite buffer. It is also shown that increasing the buffer size pushes the floor region to lower bit error rate values.

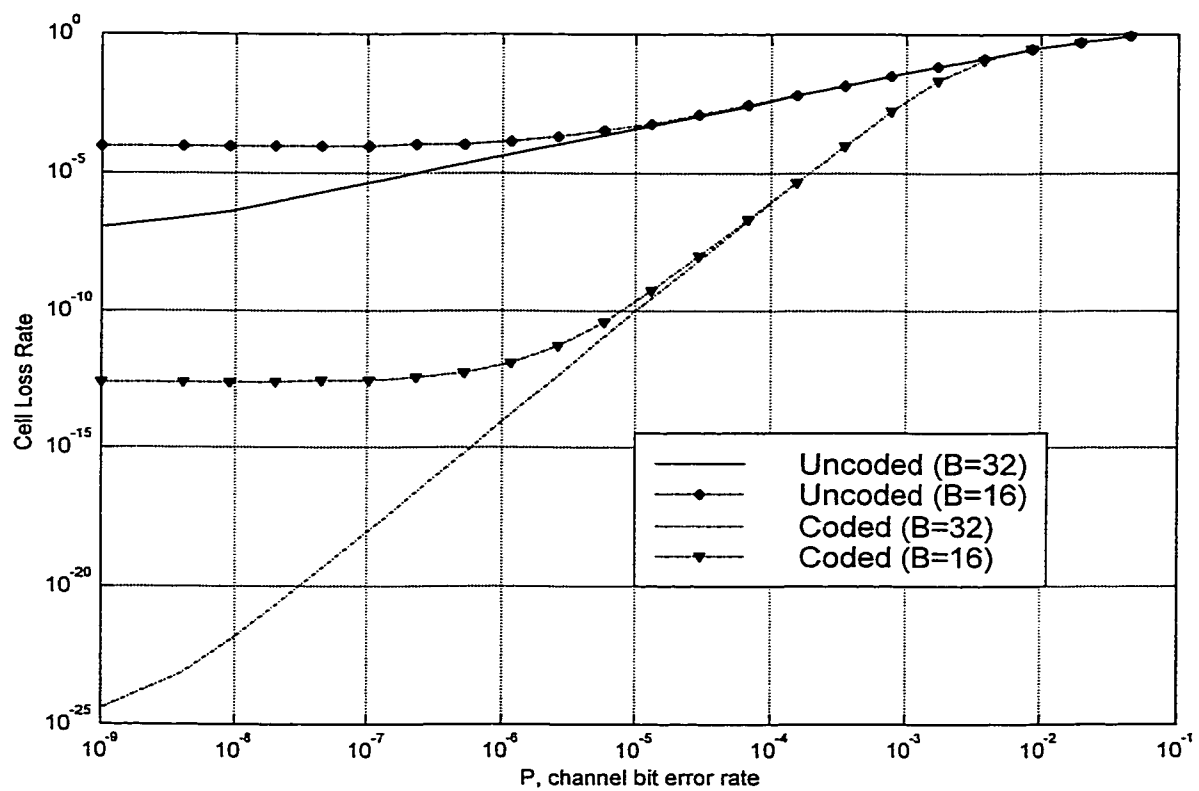


Figure 3.4: Studying the effect of different buffer sizes on cell loss performance for random channel and random traffic loading with  $\rho=0.8$

Figure 3.5 shows the effect of traffic intensity on the cell loss rate for  $B=32$ . Two values of  $\rho$  are considered  $\rho = 0.85$  and  $0.8$ . As we increase the traffic intensity, the buffer overflow probability increases and, consequently, the cell loss rate is also increased.

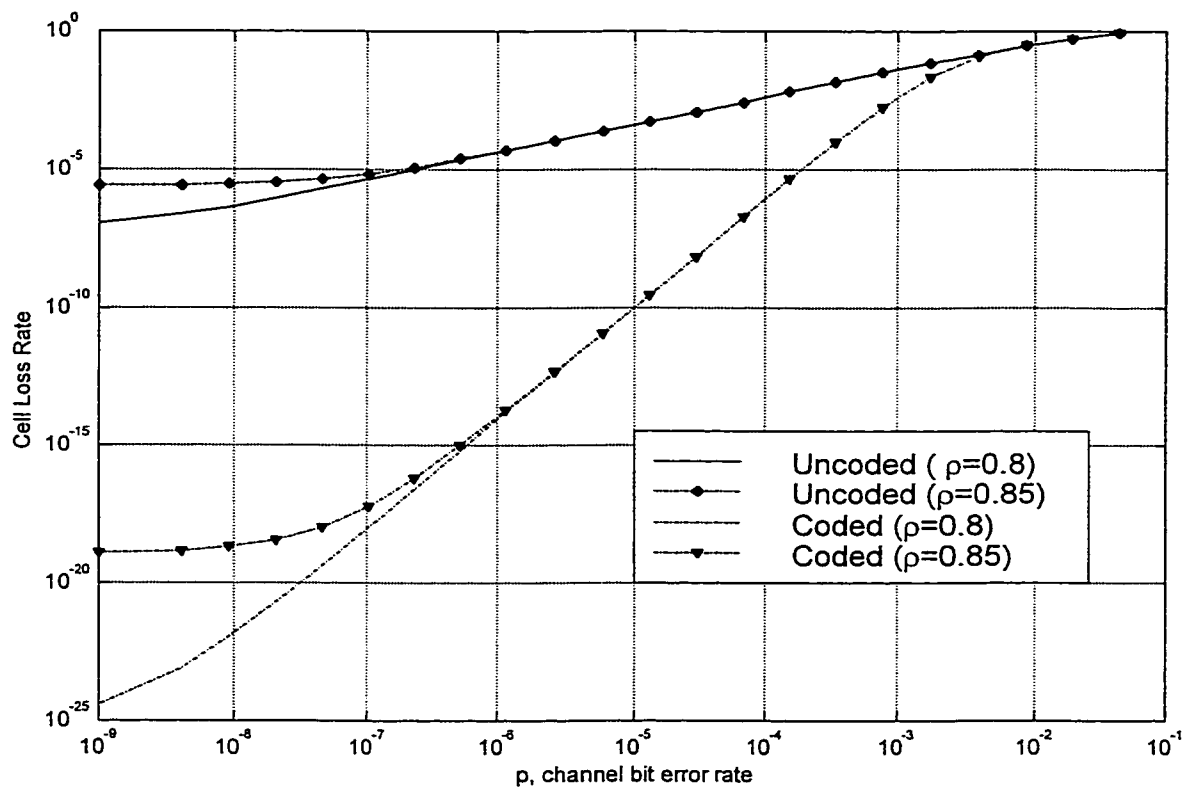


Figure 3.5: Studying the effect of different buffer sizes on cell loss rate for random channel and random traffic loading with  $B=32$

Figure 3.6 shows the cell loss rate versus the buffer size at  $p = 10^{-6}$  and for different values of traffic intensities. The figure shows that at traffic intensity of 0.8 the buffer size of  $B = 26$  is large enough to assume essentially the same performance as that of infinite buffer. The figure also shows the effect of increasing the traffic intensity from  $\rho = 0.8$  to  $\rho = 0.85$ , increasing the cell loss rate. Increasing the buffer size to 36 combats the effect of the increased traffic intensity. In other words, a buffer size of 36 provides the same effect as an infinite buffer for a traffic intensity up to 0.85.

In general, as the buffer size increases the cell loss rate decreases up to a point where a further increase in the buffer size does not improve the performance. At the point the cell loss becomes mainly due to channel errors.



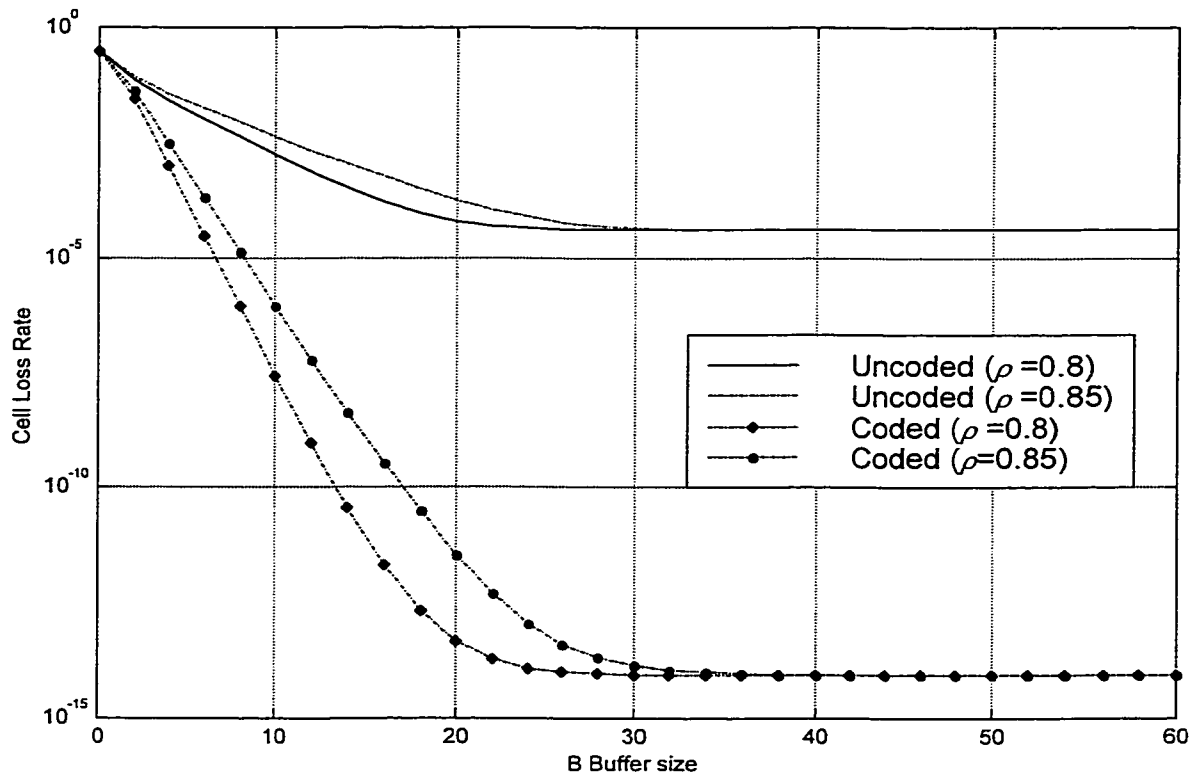


Figure 3.6: Cell loss rate versus Buffer size for random channel with  $p = 10^{-6}$

Next we study the joint effect of finite buffer size and channel correlation. Figure 3.7 shows that cell loss rate at a correlation coefficient of 0.75 and buffer size  $B=16$ . The traffic load is taken to be  $\rho = 0.8$ . The figure shows that at high bit error rate's region, where the cell loss due to channel errors is dominant, the cell loss rate for the correlated channel is smaller than that of random channel for reasons explained in the last chapter. On the other hand, when the cell loss probability due to buffer overflow dominates, at low bit error rates, the floor region effect occurs and the cell loss performance for both random and correlated channels becomes the same.

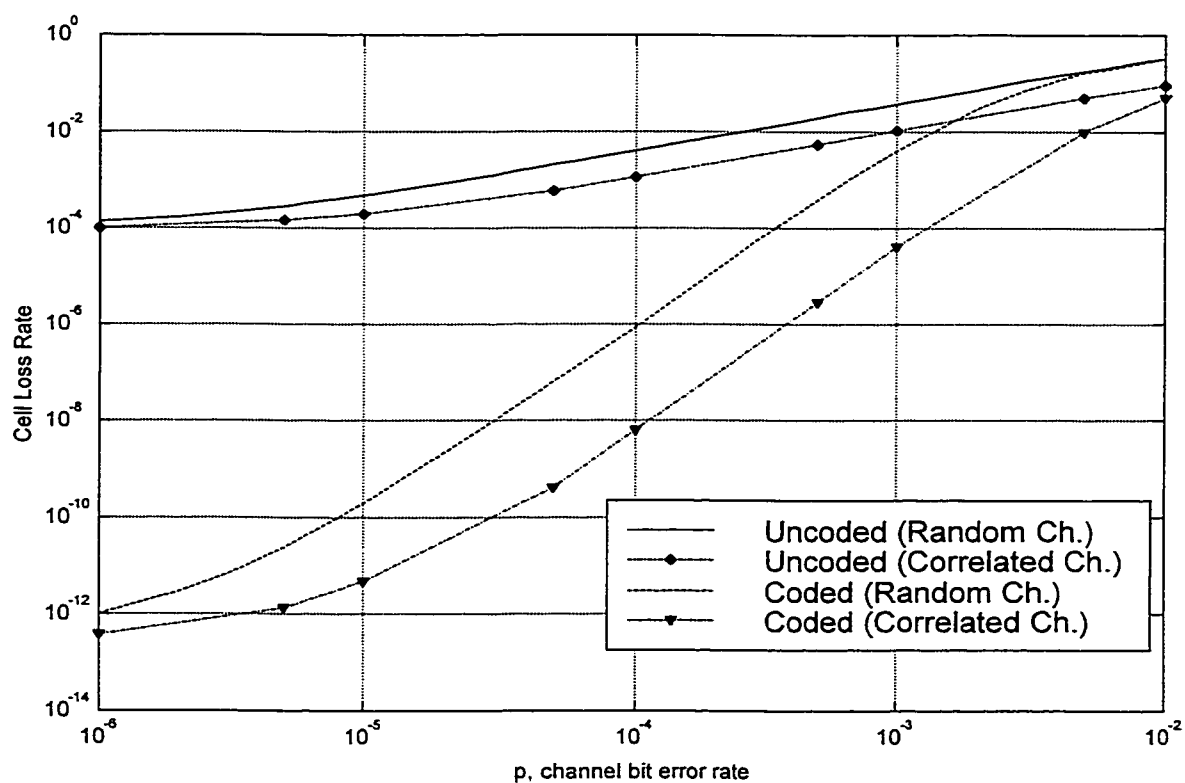


Figure 3.7: Studying the effect of Finite Buffer  $B=16$ , on correlated channel with correlation coefficient  $\omega=0.75$ , and comparison with the random channel case.

Simulation of the two–dimension matrix code including the buffer effect is performed and compared with the analytical results as shown in Figure 3.8 and Figure 3.9 for cell loss rate and cell error rate respectively. The figures are obtained for random traffic loading with traffic intensity of 0.8 and buffer size of  $B=16$ , and random channel errors. The perfect match between analytical and simulation results prove the validity of our analysis.

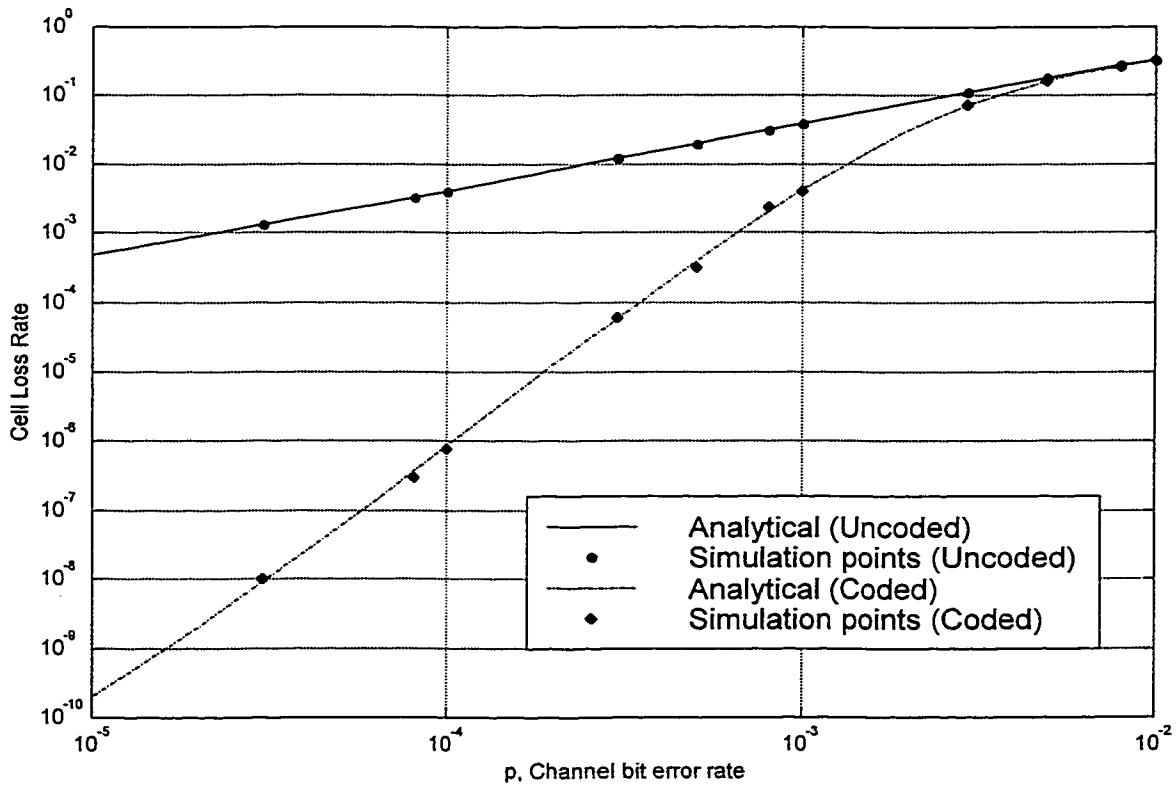


Figure 3.8: Simulation and Analytical results for random channel for Uncoded and Coded Cell loss rate with  $B=16$  and  $\rho=0.8$

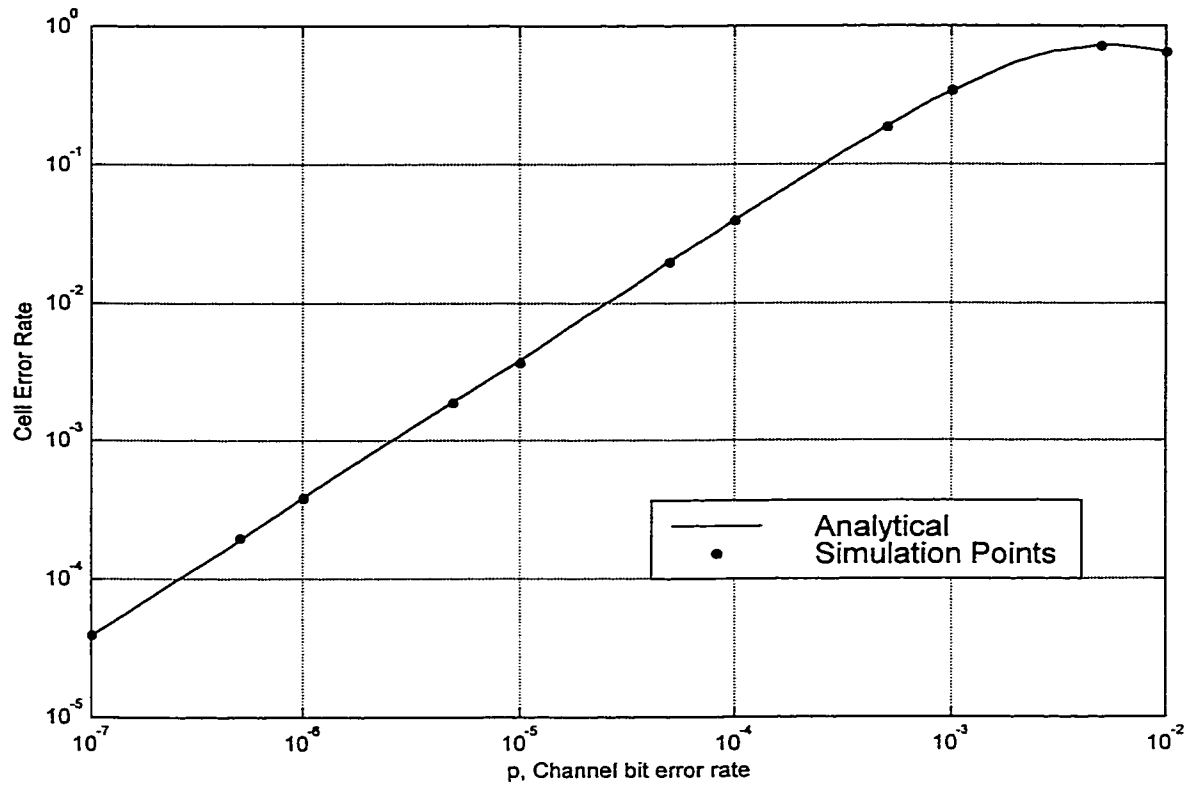


Figure 3.9: Simulation and Analytical results for random channel for Coded Cell error rate with  $B=16$  and  $\rho=0.8$

### **3.3 Bursty Traffic Loading:**

The effect of buffer overflow under bursty traffic loading, which results in burst cell loss under congested conditions, is investigated in this section. Because it is difficult to formulate the burstiness of the input traffic and to reflect the bursty nature effect on the matrix code analytically, a simulation is adopted instead.

#### **3.3.1 Input Traffic Model**

The model for the bursty traffic is adopted from [35]. Consider an  $X \times Y$  ATM switch with output queueing at the node as shown in Figure 3.1. One of the inputs is dedicated to the tagged input traffic, which interferes with  $X-1$  untagged traffic stream, and concentrating on the tagged output port, the node is modeled by an  $X$ -to-1 ATM multiplexer where the output buffer is a FIFO queue of capacity  $B$  cells shown in Figure 3.10.

We consider  $X$  independent discrete-time Markovian cell sources at the input of the multiplexer, which has identical statistics. It is assumed that the tagged source, which is coded, traverses a noisy link, while the untagged sources are coming through high quality ATM links. The two-state Markov chain of one source is shown in Figure 3.11, where states 0 and 1 correspond respectively to the “idle” and “active” states of the source.

The state transition occur once per time slot, which is the unit of time required to transmit a cell over the output link, and a source generates one cell per time slot when it is active.

The steady state behavior, described by the limiting probabilities of the “1” and “0” states, is given by [35,36]

$$\pi_0 = \frac{1-\beta}{2-\beta-\alpha} \quad (3.18)$$

$$\pi_1 = \frac{1-\alpha}{2-\alpha-\beta} = \rho_1 \quad (3.19)$$

where  $\rho_1$  is the normalized load offered by one source and  $\alpha$  and  $\beta$  are the transition probabilities within the states. Consequently, the normalized aggregate load becomes

$$\rho = \sum_{m=1}^X \rho_1 = X\rho_1 \quad (3.20)$$

since the sources are independent and identical for the uncoded systems.

When the tagged traffic is coded, the Markovian behavior of Figure 3.11 is slightly distorted due to the periodic insertion of parity cells. This disturbance is ignored here, and we assume that the Markovian behavior is preserved after encoding. Yet, we

account for the parity overhead by inflating the normalized load of tagged source, for an  $M \times N$  coded matrix size, that is

$$\rho_{et} = \frac{MN}{(M-1)(N-1)} \rho_1 \quad (3.21)$$

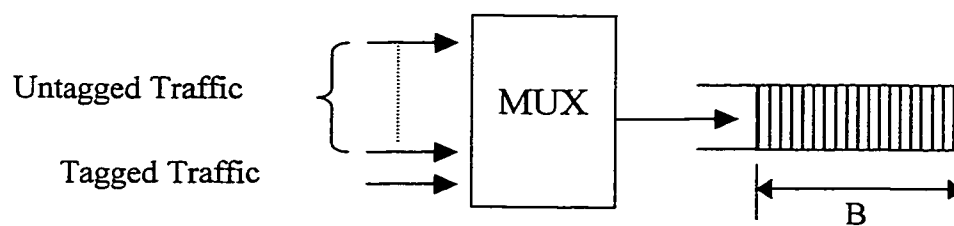


Figure 3.10: ATM node model

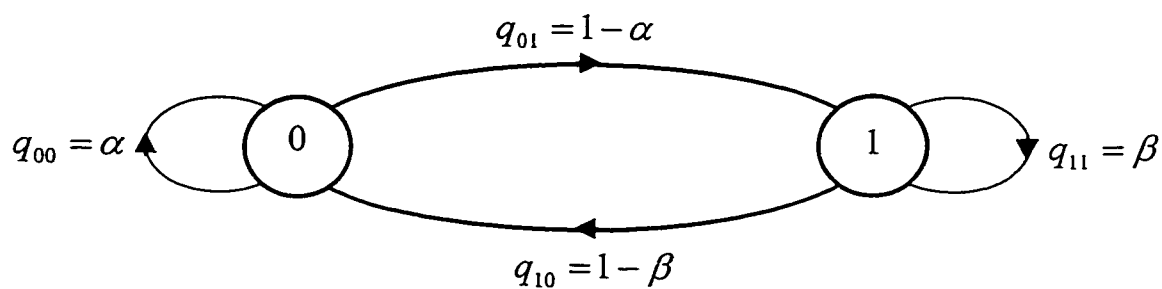


Figure 3.11: Markov Chain model of one source

The two-state Markovian source *generates* a correlated, or bursty, traffic in general. The burstiness of a source is measured by the *clustering coefficient*,  $c$ , defined as [35],

$$c = \alpha + \beta \in (0, 2) \quad (3.22)$$

In general, the closer  $c$  to 2 is, the burstier the source becomes. When  $c=1$  the process becomes an independent Bernoulli process, and a completely uncorrelated cell stream is generated, thus, the model is reduced to the random traffic loading case studied in section 3.2.

The remaining  $K = X - 1 > 1$  untagged sources are grouped to form an aggregate source, which is modeled by a  $(K + 1)$ -state discrete-time Markov Chain. The state variable is the number of active untagged sources, or equivalently, the number of simultaneous untagged cells at the output link under concentration. The state of the tagged source is denoted by  $t \in \{0, 1\}$  which indicates presence or absence of cell, while the untagged source states are denoted by  $u = \{0, 1, 2, \dots, X - 1\}$  which represents the number of active untagged sources at certain time slot.

Let  $q_{ij}, i, j = 0, 1, 2, \dots, K$ , be the state transition probabilities for untagged sources Markov Chain. The transition from state  $i$  to state  $j$  occurs when  $l$  of  $i$  active sources



become idle, and  $j - (i - l)$  of  $K - i$  idle sources become active simultaneously in a slot.

Therefore the state transition probability for the untagged source is given by

$$q_{ij} = \sum_{l=\underline{l}}^{\bar{l}} \binom{i}{l} (1 - \beta)^l \beta^{i-l} \binom{K-i}{j-i+l} (1 - \alpha)^{j-i+l} \alpha^{K-j-l} \quad (3.23)$$

where  $\underline{l} = \max(0, i - j)$  and  $\bar{l} = \min(i, K - j)$ . Note that when we set  $K=1$ , the transition probabilities in Equation (3.23) reduce to the transition probabilities in Figure 3.11.

### **3.3.2 Simulation:**

The flowchart used to simulate the performance of the matrix code under bursty traffic is given in Figure 3.12. In the initialization we define the single source parameters  $\alpha$  and  $\beta$  using Equations 3.19 and 3.20 for given aggregate traffic intensity,  $\rho$ , and clustering coefficient  $c$ . The untagged sources are superposed to form  $(K+1)$ -state Markov chain where the state transition probabilities are computed using Equation 3.23. For the tagged source the parameter  $\beta$  is modified to reflect the effect of the inflation of traffic due to the added parity cells of the matrix code.

The generated cell from the tagged source, denoted by  $t$ , is passed through the channel to the ATM switch. The cell header is checked and discarded if it contains errors. The

untagged cells,  $u$ , pass through high quality link, and therefore, the cell loss due to channel errors is assumed to be negligible.

The surviving tagged cells (from channel errors) and untagged cells are multiplexed and passed to the buffer. The multiplexer is assumed to be internally non-blocking and capable of transporting all the simultaneous input cells to the buffer in zero time. If the buffer is empty, a randomly chosen cell of the new cells is transmitted immediately over the output link, and the rest is stored in a random order. Otherwise, the leading cell in the buffer is transmitted first, and the new cells are stored in a random order, provided that there is enough room for them. Therefore, the buffer occupancy (the number of cells in a time slot) is characterized by the relation,

$$b_s = \min(B, \max(0, b_{s-1} + tr_s + ur_s - 1)) \quad (3.25)$$

where the index  $s$  is the slot variable,  $tr \in \{0, 1\}$  is the state of the tagged cell that survives after checking the header for channel errors,  $ur = u \in \{0, 1, 2, \dots, X-1\}$  is the number of cells of untagged sources, and  $b \in \{0, 1, 2, \dots, B\}$  is the buffer occupancy after  $tr + ur$  new cells arrive in a slot.

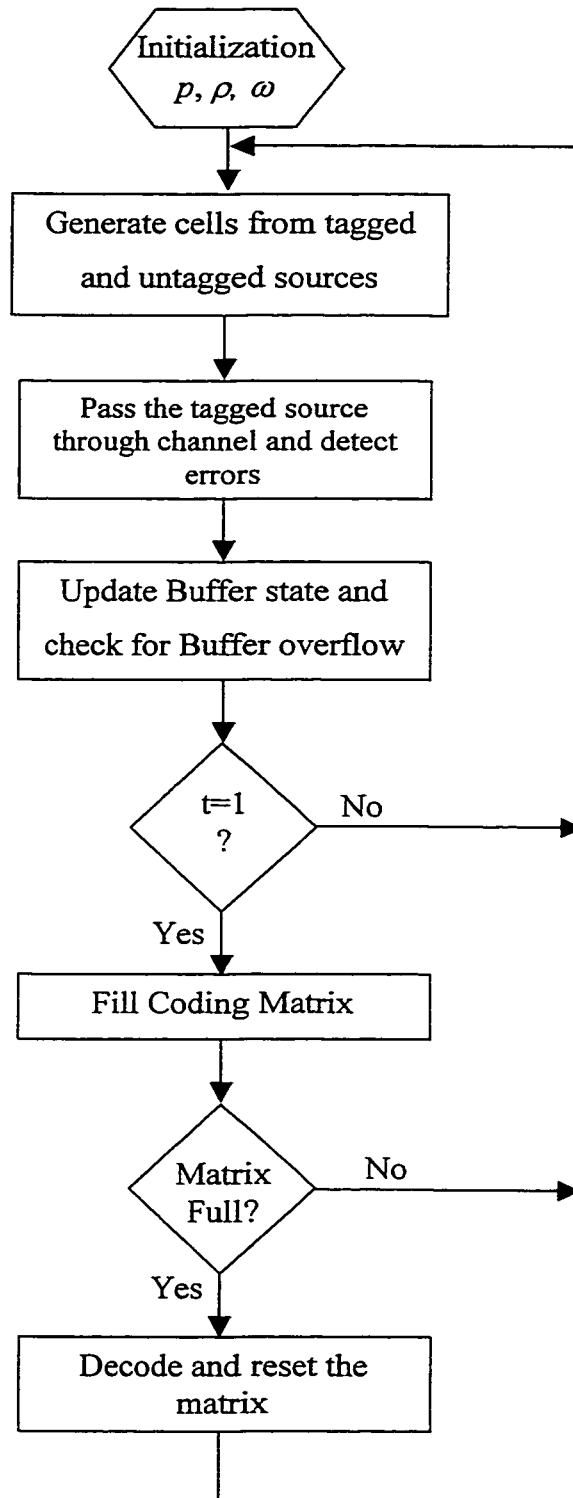


Figure 3.12: Flow chart for Simulation the performance of two-dimension code with bursty traffic

The maximum number of new cells that can be served in the slot  $s$  is  $B - b_s + 1$ . Therefore, if there are  $tr_s + ur_s > B - b_s + 1$  new cell arrivals, a randomly chosen set of  $tr_s + ur_s - (B - b_s + 1)$  of them will be discarded. In particular if  $tr_s = 1$ , that is a tagged cell is present and has survived channel errors, and  $ur_s > B - b_s$ , then the tagged cell will be discarded with probability  $1 - (B - b_s + 1)/(1 + ur_s)$ , and the buffer becomes full.

The tagged cells that survive both channel errors and buffer overflow are restructured in the coding matrix, where the locations of lost cells are filled with all-zero cells. Once the code matrix is formed, decoding is performed in the usual manner.

### **3.3.3 Discussion and Results for Random Channel:**

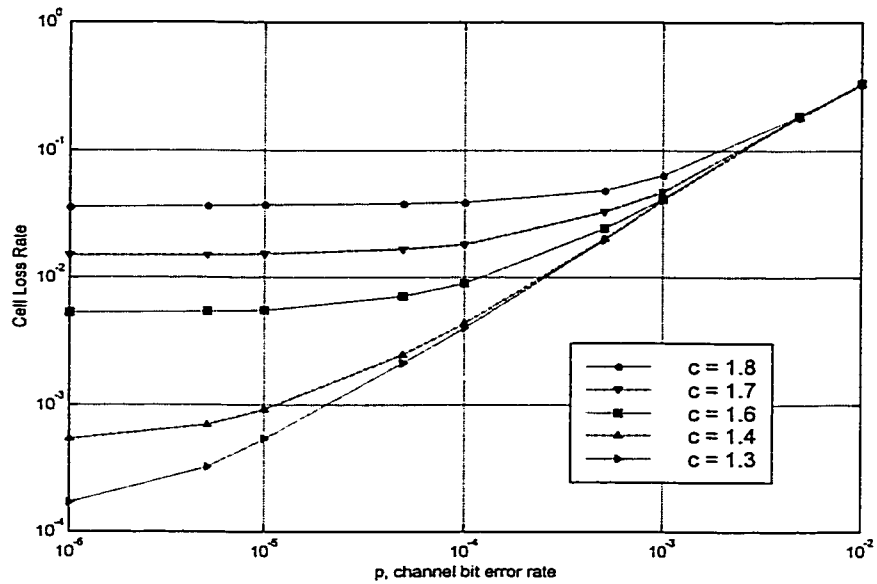
To study the effect of burstiness of traffic on the code performance simulation is performed with different values of clustering coefficient  $c$ . The buffer size is set to  $B=32$  and the aggregate traffic intensity is set to  $\rho = 0.8$ . Figure 3.13 shows the cell loss rate versus channel bit error rate for random channel errors with different clustering coefficient  $c$ ; the uncoded is depicted in Figure 3.13 (a), where as coded system is depicted in Figure 3.13 (b). The effect of the burstiness is clear on the buffer overflow, which is reflected on the code performance. For a high clustering coefficient, such as  $c=1.8$ , cells are transmitted in long bursts. Therefore, when a source at the input of the ATM switch is transmitting a cell to a certain output link, it is most likely to transmit cells during the consecutive slots to the same output link. And having different sources

transmitting to the same output at the same time slot cause the buffer to be occupied by a large number of cells during consecutive time slots, and thus the buffer overflow becomes more frequent.

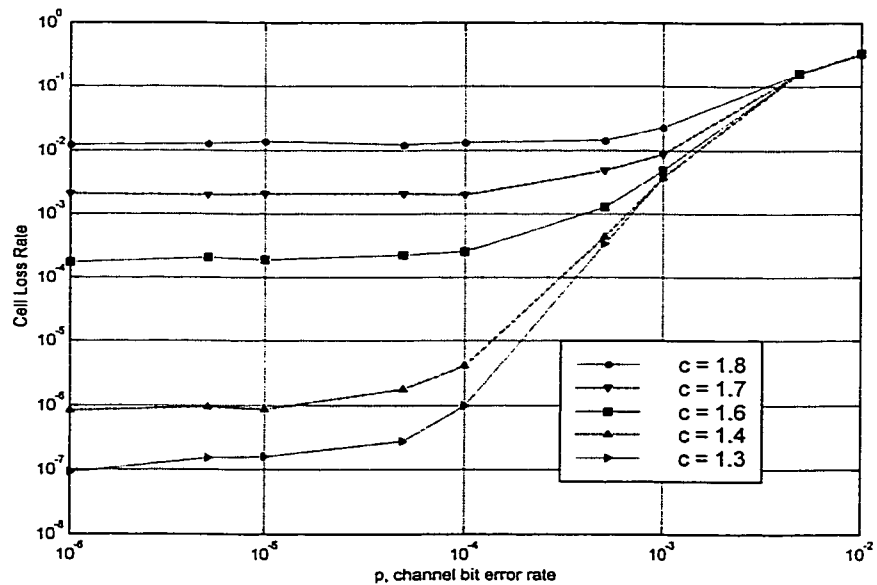
The figure shows that the code performance is poor for high bursty traffic, and for such traffic cell loss during the congestion periods occur in bursts. These bursts may be longer than the row size of the decoding matrix. It is also possible that two or more shorter bursts take place within the same decoding matrix. Moreover, there are those cells lost due to channel errors. As a result of all these scenarios, it is highly probable to find more than one lost cell, in columns and rows, and are not recoverable.

When the burstiness of the source is decreased  $c=1.6$ , the cells are transmitted in shorter bursts and thus the buffer occupancy by certain source is decreased, which cause the congestion to occur less frequent. This will cause lesser cells to be lost and shorter bursts to occur within decoding matrix, which allow more cells to be recovered and thus improving the code performance.

In general, as the burstiness of the input traffic decreases the buffer overflow probability is reduced and the burst length gets smaller, which makes the code more effective.



(a)



(b)

Figure 3.13: Study of source burstiness effect on cell loss performance with buffer size  $B=32$  and aggregate traffic intensity  $\rho=0.8$ : (a) Uncoded system; (b) Coded system

To demonstrate that the model under study reduces to the model adopted for random traffic loading, simulation is performed by setting  $c=1.0$ , the case of uncorrelated input traffic. The buffer size was set  $B=32$  and the aggregate traffic intensity to  $\rho = 0.8$ . The analytical and simulation results for cell loss rate are shown in Figure 3.14, respectively. The figure shows that the simulation results are identical to the results obtained analytically.

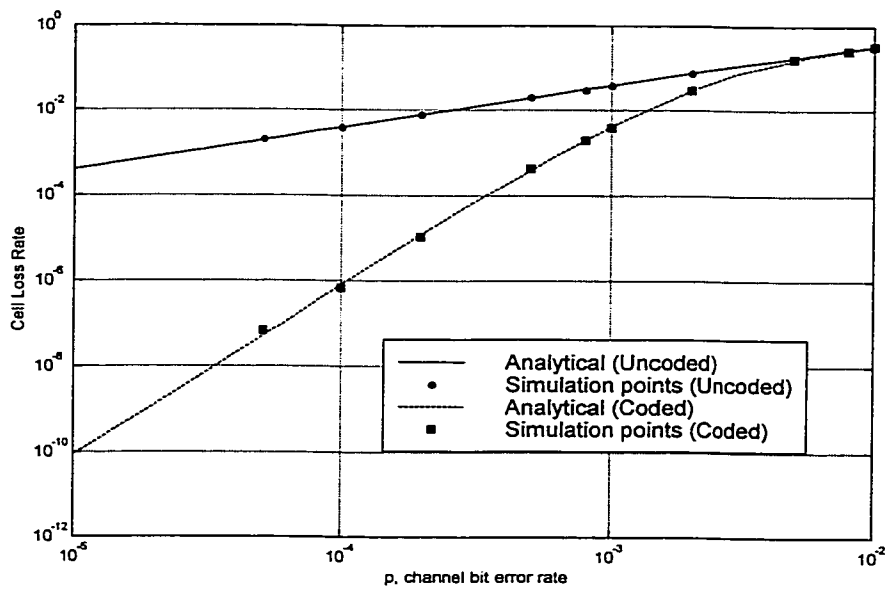
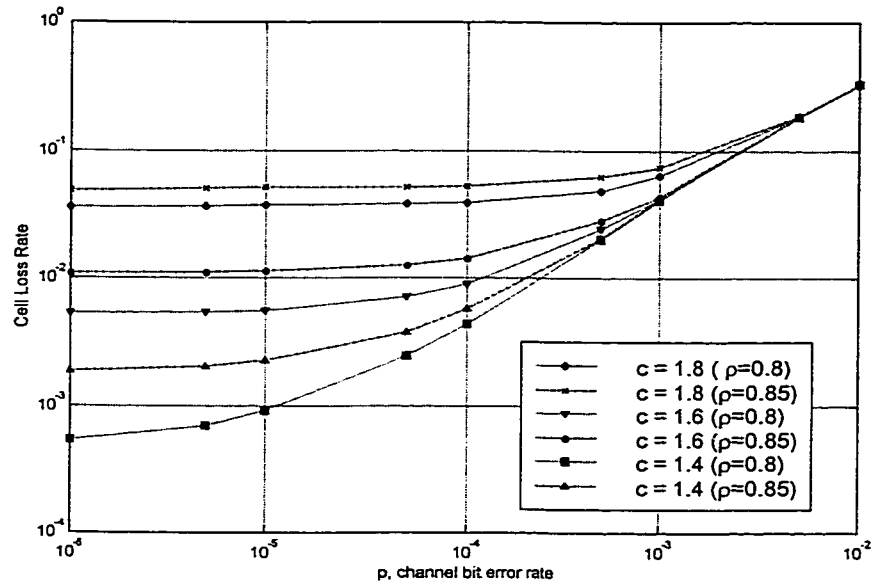


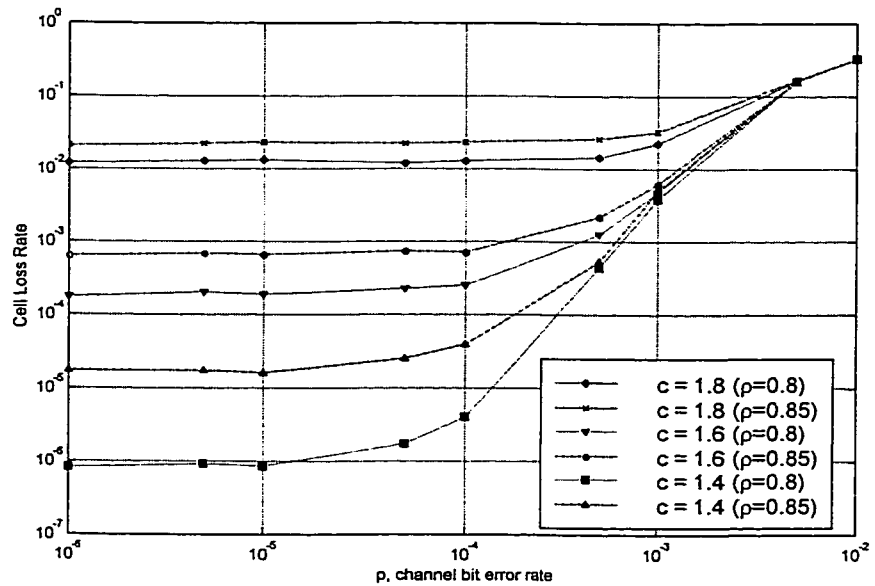
Figure 3.14: Cell loss performance comparison between random traffic loading (Analytical) and bursty traffic loading with clustering coefficient  $c=1.0$

The effect of traffic intensity on cell loss rate is simulated and plotted in Figure 3.15. Figure 3.15 (a) shows the probabilities for the uncoded system, while Figure 3.15 (b) shows the probabilities for the coded system. Aggregate traffic intensities of  $\rho = 0.8$  and  $\rho = 0.85$  at various clustering coefficients are investigated. The buffer size was set to  $B=32$ . The figure shows that increasing the aggregate traffic intensity from  $\rho = 0.8$  to  $\rho = 0.85$  increases the burst length and hence increases the buffer overflow probability. This, in effect, causes the cell loss rate to increase. The figure also shows the effect of traffic burstiness for different traffic intensities.





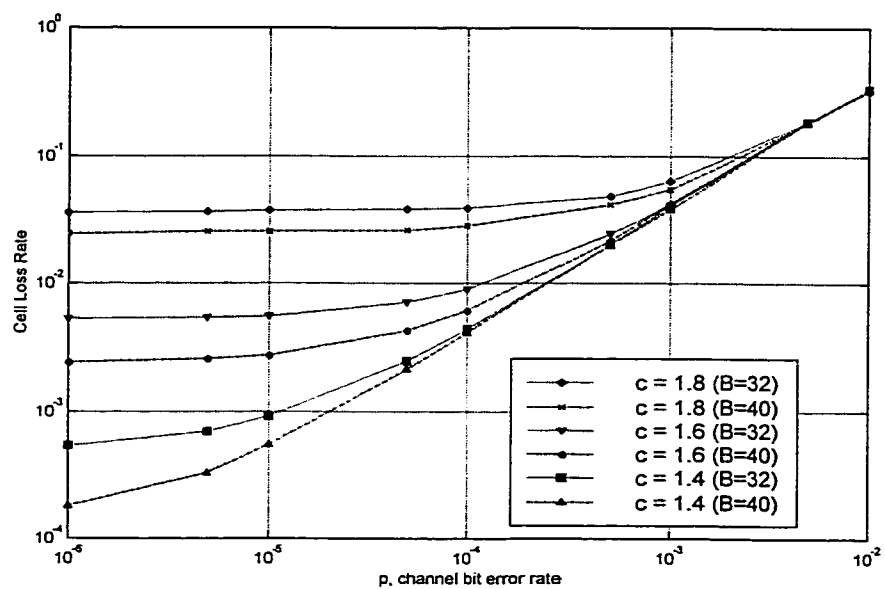
(a)



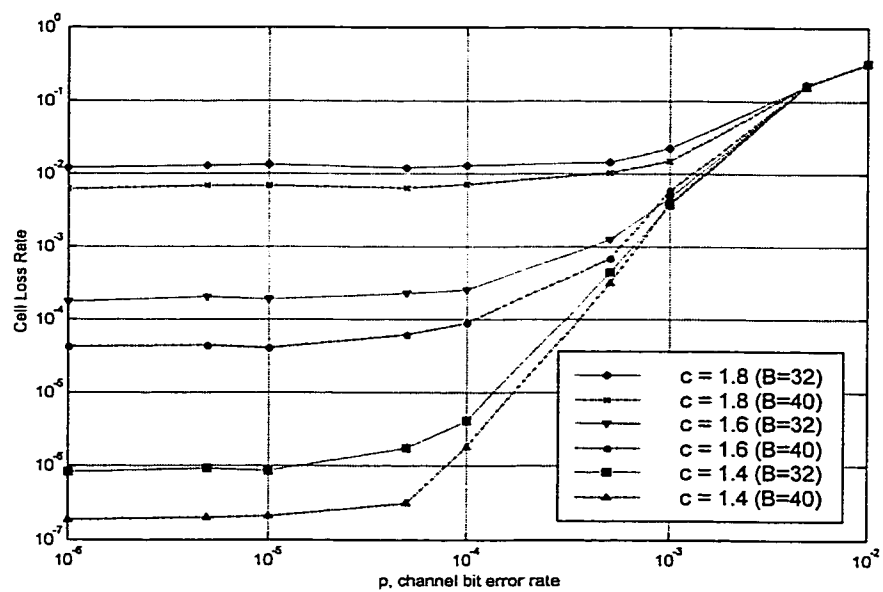
(b)

Figure 3.15: Effect of aggregate traffic intensity on cell loss performance for random channel with buffer size  $B=32$ : (a) Uncoded cell loss rate; (b) Coded cell loss rate

Figures (3.16, 3.17) study the effect of buffer size on cell loss rate. Figure 3.16 shows the cell loss rate for two buffer sizes, namely 32 and 40, at an aggregate traffic intensity of 0.8, while Figure 3.17 shows the same for  $\rho = 0.85$ . The figures show significant reduction in cell loss rate by increasing the buffer size. The explanation is straight forward; as the buffer size increases, it gives capacity for more cells and thus for given clustering coefficient it reduces the probability that a cell will be lost due to buffer overflow. Although increasing the buffer size has improved the performance, the buffer overflow probability is still the dominant factor for cell loss, mainly because of the burstiness nature of the input source.

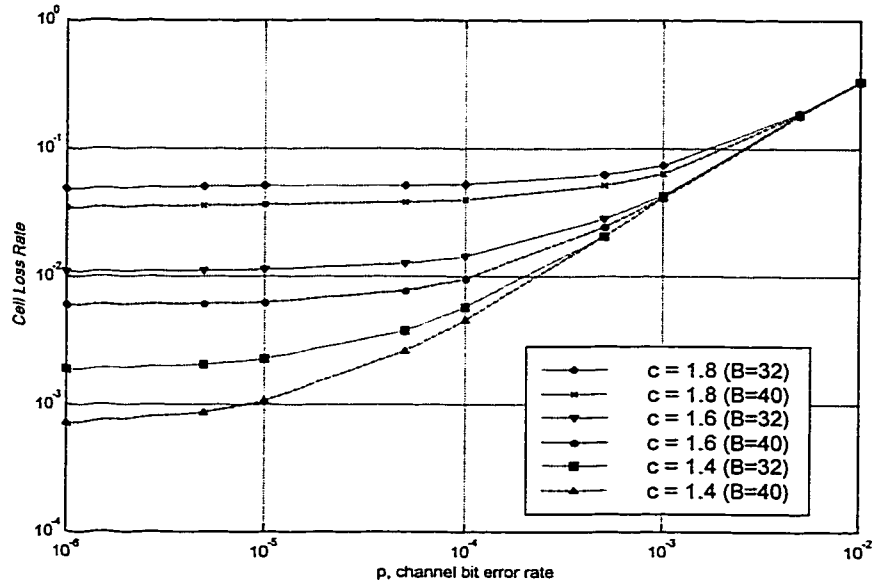


(a)

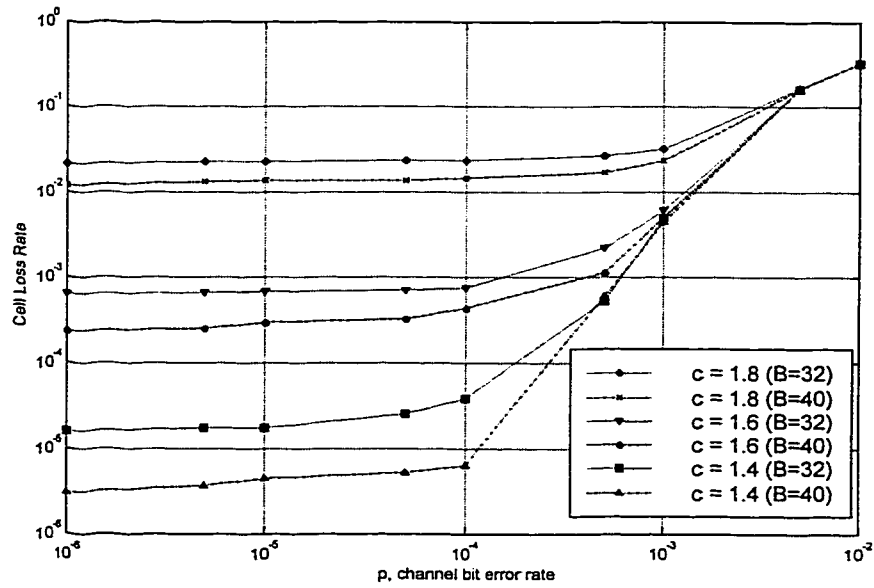


(b)

Figure 3.16: Study of Buffer size effect on cell loss rate for random channel with aggregate traffic intensity  $\rho=0.8$ : (a) Uncoded system; (b) Coded system



(a)



(b)

Figure 3.17: Study of Buffer size effect on cell loss rate for random channel with aggregate traffic intensity  $\rho=0.85$ : (a) Uncoded system; (b) Coded system

Cell loss rate versus buffer size versus for random channel with bit error rate  $p = 10^{-4}$  and a source with clustering coefficient  $c=1.4$  is shown in Figure 3.18, while it is shown for random traffic loading in Figure 3.19. The figures show the uncoded and coded cell loss rate for two different values of aggregate traffic intensity  $\rho = 0.8$  and  $\rho = 0.85$ . Very significant results can be drawn from these figures. Consider the curves in both figures corresponding to the traffic load of 0.8. For the random traffic case, it is observed that a buffer size of 20 cells is large enough to accommodate all incoming cells, where as buffer size of at least 60 cells is needed to reach this stage for bursty traffic with clustering coefficient of 1.4. Note that both cases the cell loss rate approach the same floor ( $\approx 10^{-6}$ ) at  $\rho = 0.8$ , which is a limit set by the channel bit error rate. The same observation can be made for  $\rho = 0.85$  at buffer size of 24 & 80. In other words, there is always a buffer size value above which the operation of the buffer approaches that of infinite buffer. This value is a function of the traffic load, but more importantly the bursty nature of the traffic.

Also the figure shows that although the uncoded cell loss rates for both  $\rho = 0.8$  and  $\rho = 0.85$  converge to the same at a buffer size of 50, the coded cell loss rates do not. The difference is due to the burst nature of the lost cells. In the case of  $\rho = 0.85$  the bursts are longer. Consequently, the decoding becomes less effective which results in a higher cell loss rate.

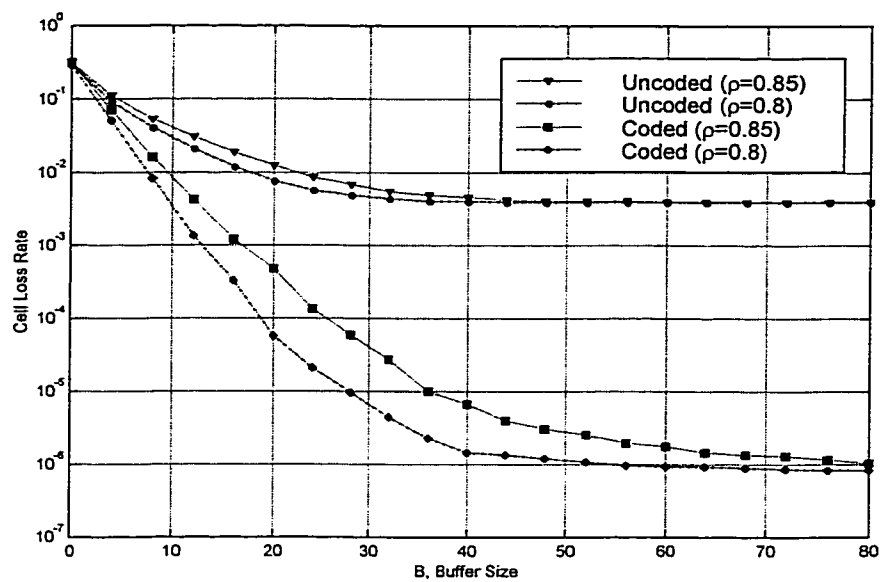


Figure 3.18: Cell loss rate versus Buffer size for bursty source with clustering coefficient  $c=1.4$ .

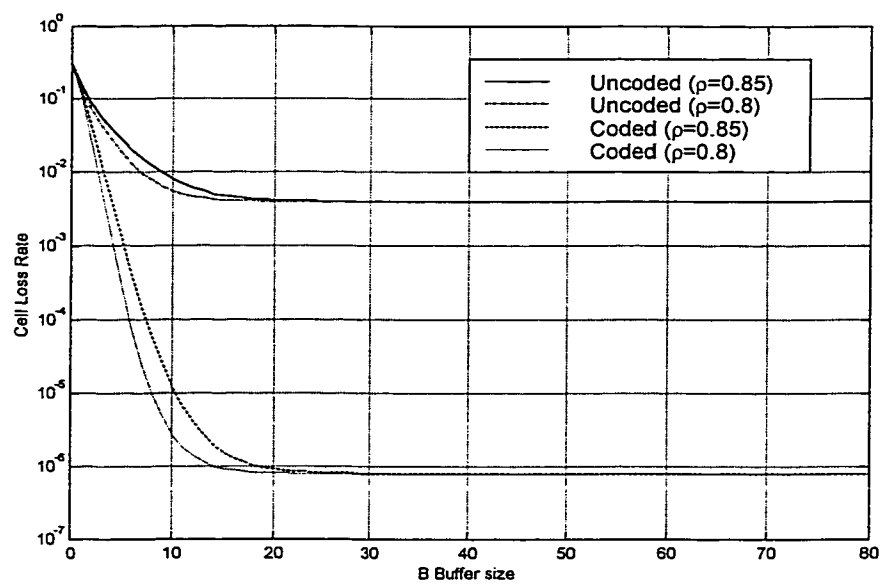


Figure 3.19: Cell loss rate versus Buffer size for random traffic loading

Cell loss rate versus clustering coefficient,  $c$ , with bit error rate  $p = 10^{-4}$  and a source with aggregate traffic intensity  $\rho = 0.8$  is shown in Figure 3.20. The figure shows the uncoded and coded cell loss rate for different buffer sizes  $B=32$  and  $B=40$ . As mentioned before the effect of input traffic burstiness is apparent, i.e. when the input traffic burstiness is high the cell loss performance is poor and as the burstiness decreases the cell loss rate is decreased. Also, as the burstiness measure decreases the cell loss performance approaches to the cell loss performance with random traffic loading.

The effect of the buffer size is also shown in the figure, thus increasing the buffer size improves the cell loss performance. Note that when the burstiness measure is low the cell loss performance with both buffer sizes becomes the same. This is with the results in Figure 3.19 where we observed that any buffer larger than 24 cell approaches the performance of the infinite buffer.

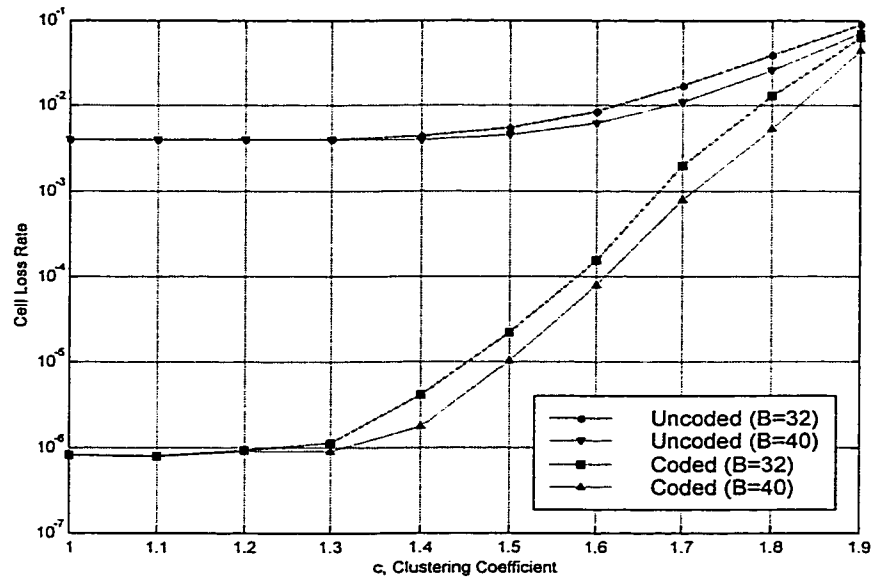
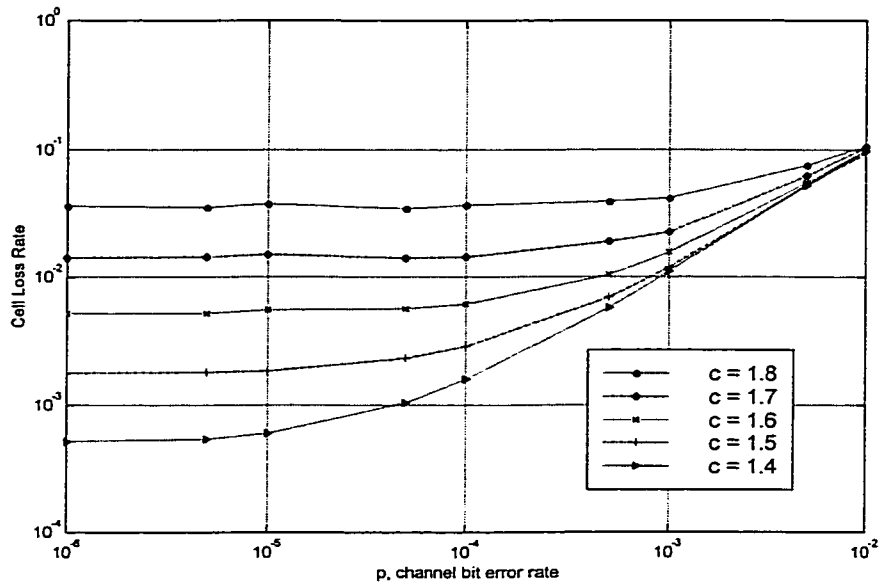


Figure 3.20: Cell loss rate versus clustering coefficient  $c$ , for random channel and aggregate traffic intensity  $\rho=0.8$

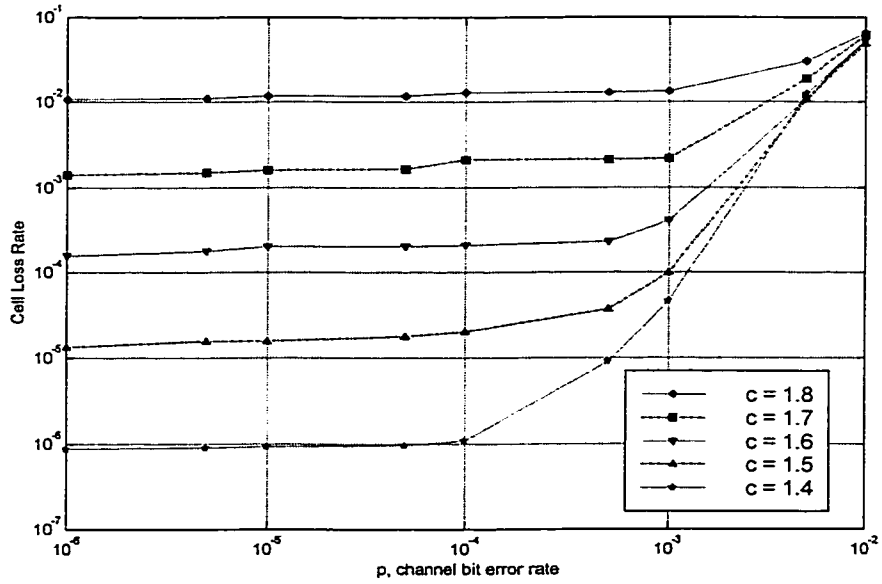


### **3.3.4 Discussion and Results for Correlated Channel:**

The same study performed on random channel is performed with correlated channel, with correlation coefficient  $\omega=0.75$ . Figure 3.21 shows the effect of traffic burstiness on uncoded and coded cell loss probabilities for correlated channel, the buffer size was set to  $B=32$  and the aggregate traffic intensity was  $\rho = 0.8$ .



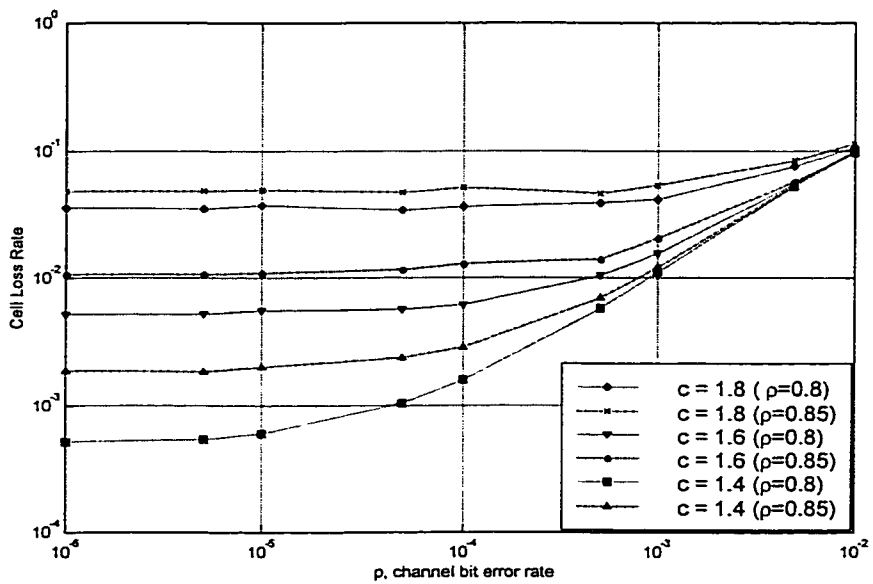
(a)



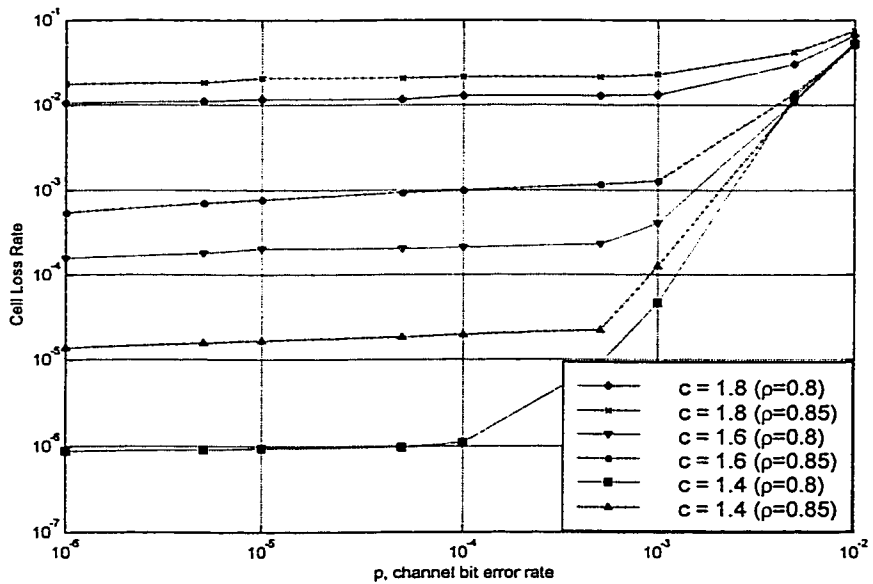
(b)

Figure 3.21: Effect of traffic burstiness on cell loss rate with buffer size  $B=32$  and aggregate traffic intensity  $\rho=0.8$ : (a) Uncoded system; (b) Coded system

The effect of aggregate traffic intensity on the cell loss performance for correlated channel is shown in Figure 3.22. The buffer size was set to  $B=32$  and different values for burstiness measure were used as indicated in the figure.



(a)

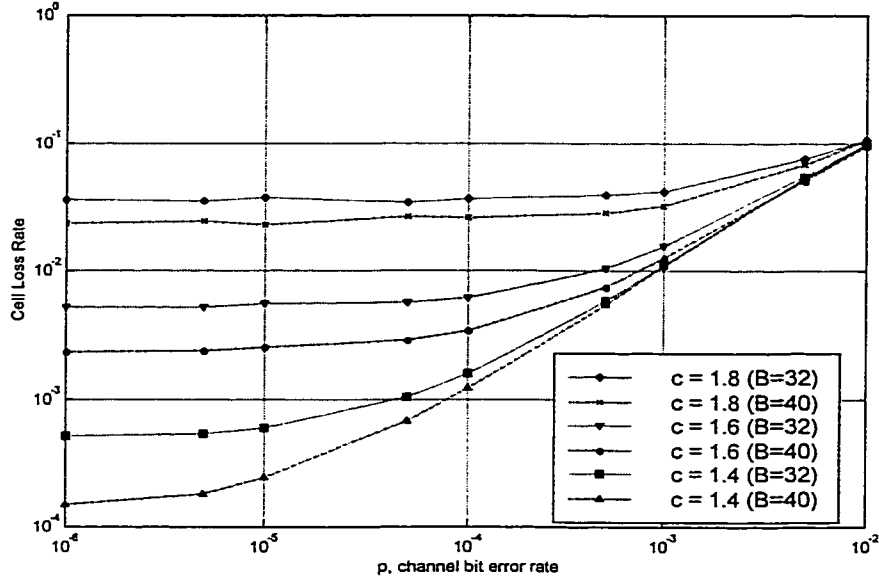


(b)

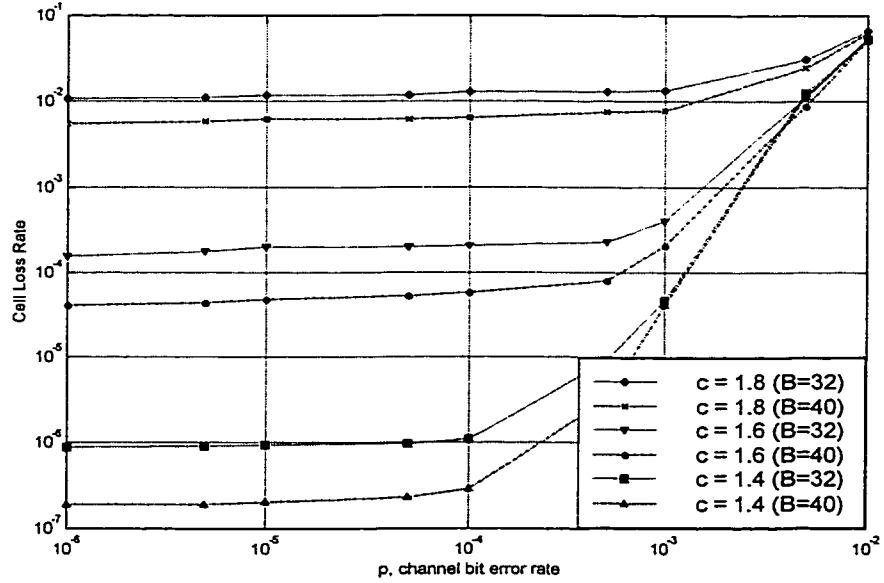
Figure 3.22: Effect of aggregate traffic intensity on cell loss rate with buffer size  $B=32$ : (a) Uncoded system; (b) Coded system

The effect of Buffer size on correlated channel is plotted in Figures (3.23 and 3.24) with different burstiness measure. Figure 3.23 shows the buffer size effect with aggregate traffic intensity  $\rho = 0.8$ , while Figure 3.24 shows it with aggregate traffic intensity  $\rho = 0.85$ .

It is seen that all observation and conclusions made for the random channel case do apply here.

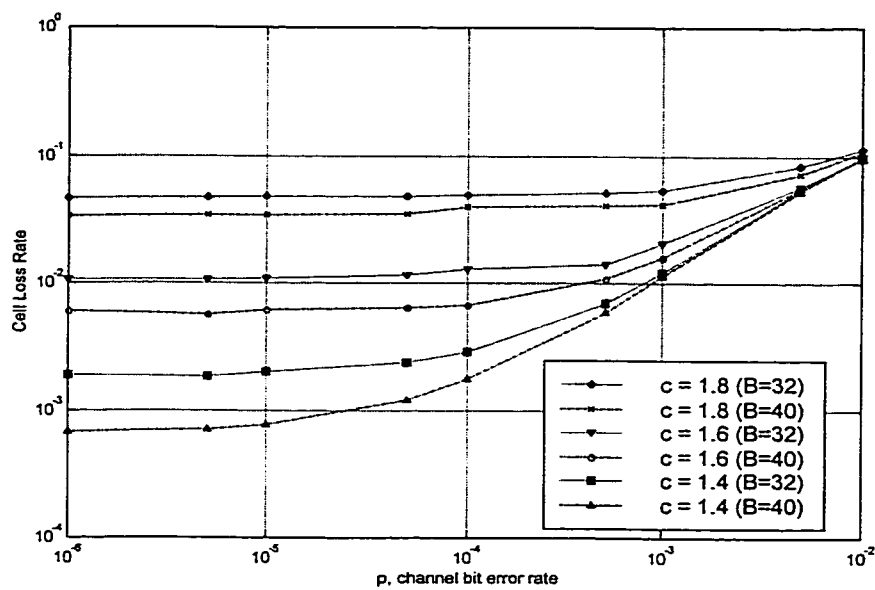


(a)

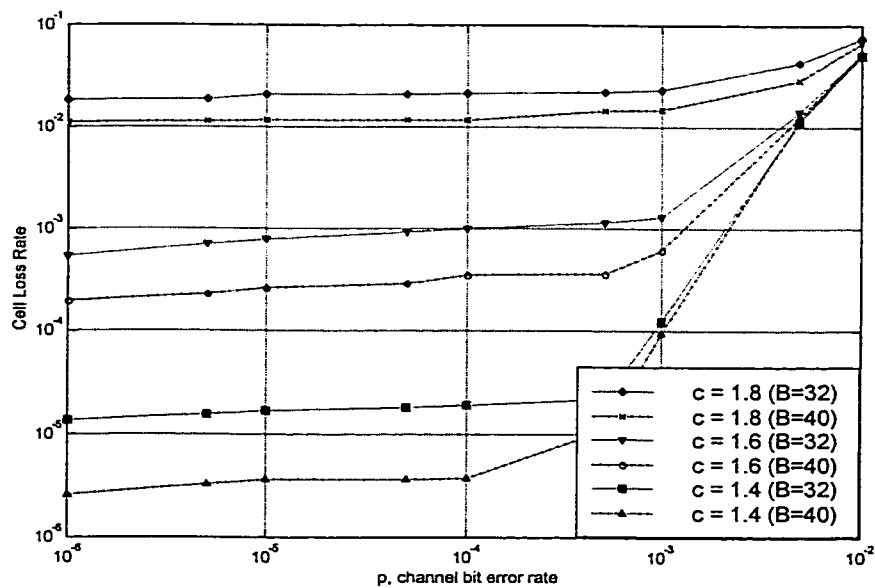


(b)

Figure 3.23: Effect of Buffer size on cell loss rate for bursty sources with aggregate traffic intensity  $\rho=0.8$ : (a) Uncoded system; (b) Coded system.



(a)



(b)

Figure 3.24: Effect of Buffer size on cell loss rate for bursty sources with aggregate traffic intensity  $\rho=0.85$ : (a) Uncoded system; (b) Coded system

The cell loss rates for random and correlated channel with bursty traffic loading are compared in Figure 3.25. The performance is plotted for clustering coefficient  $c=1.4$ , aggregate traffic intensity  $\rho = 0.8$ , and buffer size  $B=40$ . The figure shows that the cell loss rate for the correlated channel is smaller than that for random channel over the range of higher bit error rate (when the channel errors dominate the overflow probability). At low bit error rates, both channels perform essentially the same. Note that although the uncoded probabilities for the correlated channel is lower than those of the random channel at  $p = 10^{-5}$ , the coded cell loss rates are same for both channels. This is because the code performance is limited by the bursty nature of the traffic.



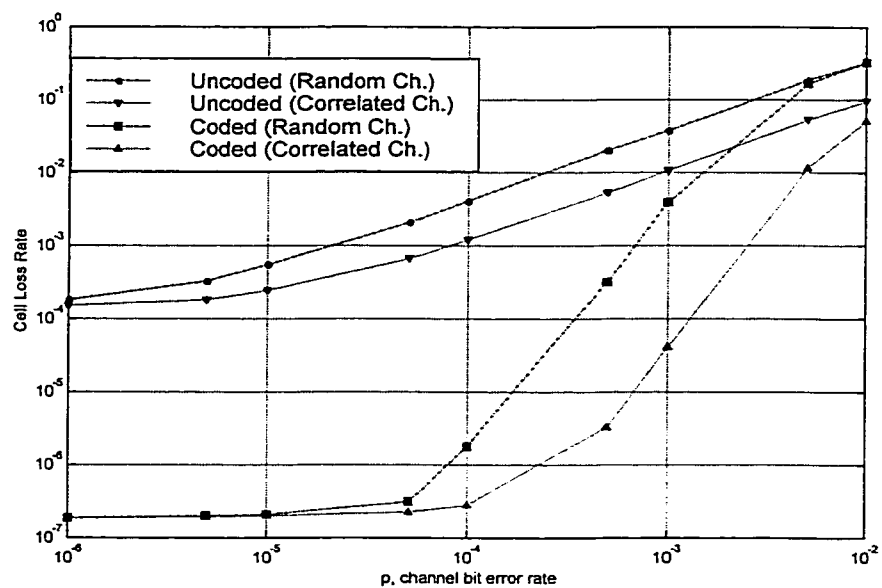


Figure 3.25: Comparison between random and correlated channel effect on cell loss performance under bursty traffic loading with clustering coefficient  $c=1.4$  and aggregate traffic intensity  $\rho=0.8$ , and buffer size  $B=40$

### **3.4 Summary:**

In this chapter, we studied the effect of finite buffer size on the two-dimensional matrix code. First, we considered the effect of random traffic loading, which causes random cells to be lost due to buffer overflow. Expressions for cell error rate and cell loss rate were derived. Then we have investigated by simulation the effect of bursty traffic loading on the code performance. The study of effect of finite buffer size was carried for both random and correlated channels. A summary of our findings is provided in the following chapter.

# CHAPTER 4

## CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

In this work we have considered a single-parity two-dimensional matrix code scheme for lost cell recovery in ATM Networks. The performance of the code is studied over a noisy ATM link. Two reliability measures were adopted: cell loss rate and cell error rate. First we have considered the effect of cell loss due to channel errors only on the performance. Under random channel errors an analytical expression was derived for the cell loss rate and cell error rate. The case of correlated channel was considered next. Simulation has been performed to evaluate the cell pre-decoding probabilities since there is no closed analytical expression for it. A Two-state Markov source was used to model a channel of a certain bit error rate and correlation coefficient.

The cell loss due to finite buffer size and its effect on code performance was considered in Chapter 3. Both the effect of random traffic loading and bursty traffic loading have been handled. In the case of random traffic loading the cell discard due to buffer overflow is random and assumed to be independent from cell loss due to channel errors. The pre-decoding probabilities of cell loss derived in Chapter 2 were modified to reflect the effect of buffer overflow and used to evaluate the code performance. With bursty traffic loading the cell loss due to buffer overflow occur in bursts. Simulation is carried out to reflect the effect of burstiness nature of cell loss on code performance. Two-state Markovian chain was used to model bursty sources. The interactive effects of source parameters (clustering coefficient and normalized traffic intensity) and buffer size were investigated.

The code which was shown to be powerful for random channel, infinite buffer case by previous work, is shown to be also powerful for correlated channel, finite buffer under random and bursty traffic. The performance over correlated channel was found to be better than that over random channel. This is because with correlated channel the errors occur in bursts, causing lesser cells to be erroneous and lost. It was also noted that in the case of finite buffer and at low channel bit error rates, when the loss due to buffer overflow becomes dominant, the performance of correlated channel and random channels becomes the same.

The code does not affect, particularly at small bit error rates, the cell error rate. This gives an indication that most of the recovered cells are recovered correctly.

It was found that for high bursty traffic the cell loss performance is severely hit and the reliability of the code is very low. As the burstiness of the incoming traffic is reduced the code performance becomes more effective in recovering larger amount of cells. Increasing the buffer size improves the performance and helps to withstand higher bursty sources.

In the finite buffer case, a finite buffer size value that gives performance equivalent to that of infinite buffer case was found. For this value the cell loss rate is limited by channel errors. It was found that for burstier traffic the buffer size value, which gives performance of infinite buffer case, is larger than that required for random traffic loading.

**Suggestion for further work:**

In view of the findings of this work, we would like to make the following suggestions for carrying out further work in this area:

1. In this study, it is assumed that the mechanism of lost cell detection is available and did not consider its effect in the analysis. The effect of Cell Loss Detecting cells on the code performance can be investigated.

2. Studying the code for matrix size larger than the  $16 \times 16$  matrix used in this study.  
This requires introducing more advanced method for cell loss detection.
3. The use of more powerful parity check on rows and columns can be investigated instead of simple parity check, especially for the case of high bursty traffic loading.
4. Study the effect of payload checking on code performance, hence increasing the reliability of the system.
5. Invoking ARQ mechanism to increase the reliability of the system by retransmitting the cells that can not be recovered is another way to extend the work in this thesis.

## References

- [1] S. Veniers, J. D. Angelepoulos, and G.I. Stassinapoulse, "Efficient Use of Protocol Stacks for LAN/MAN-ATM Internetworking," *IEEE Journal on Selected Areas in Commun.*, JSAC, vol. 11, No. 8, pp. 1160-1171, Oct. 1993.
- [2] IBM Redbook, *Asynchronous transfer mode (ATM) Technical overview*, 1995.
- [3] S. Tanenbaum, *Computer Networks*, Prentice Hall, New Jersey, 1996.
- [4] T. Chen and S. Liu, *ATM Switching Systems*, Artech House, Boston, MA, 1995.
- [5] M. de Prycker, *Asynchronous Transfer Mode: Solution for broadband ISDN*, Ellis Horwood, London, 1991.
- [6] Raif. O. Onvural, *Asynchronous Transfer Mode: Performance issue*, Artech House Inc., 1995.
- [7] J. Walrand and P. Varaiya, *High-Performance Communication Networks*, Morgan Kaufmann Publishers Inc., California 1996
- [8] M. Murata, "Requirement on ATM Switch Architectures for Quality-of-Service Guarantees," *IEICE Trans. Commun.*, vol. E81-B, No. 2, February 1998.
- [9] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, second edition, New Jersey, 1992.
- [10] G. C. Clark and J. B. Cain, *Error-Correction Coding for Digital Communications*, N.Y.: Plenum Press 1981.
- [11] Shu Lin and D. Castello, *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, N.J.: Prentice-Hall 1983.
- [12] A. M. Michelson and A. H. Levesque, *Error-Control Techniques for Digital Communication*, John Wiley and Sons Inc., New York, 1985
- [13] J. G. Proakis, *Digital Communications*, New York: McGraw-Hill, 1983.
- [14] S. B. Wicker, *Error Control Systems*, Englewood Cliffs, N.J.: Prentice-Hall Inc., 1995.
- [15] M. Y. Rhee, *Error Correcting Coding Theory*, McGraw-Hill 1989.

- [16] Peter Sweeney, *Error Control Coding: An introduction*. Englewood Cliffs, N.J: Prentice-Hall 1991.
- [17] C. G. Ommidyar, and K. Pahiavan, "Introduction to Mobile and Wireless ATM", *IEEE Communications Magazine*, vol.35, No. 11, pp. 30-33, November 1997.
- [18] P.R. Denz and A.A. Nilsson, "Performance of Error Control Coding Techniques for Wireless ATM," *IEEE ICC'98*, vol.2, pp. 1099-1103, 1998
- [19] J.B. Cain and D.N. McGreoge, "A Recommended Error control Architecture for ATM Netorks with Wireless Links," *IEEE Journal On Selected Areas In Comm.*, vol. 15, No. 1, pp. 16-28, January 1997.
- [20] Y. A. Tesafi and S.G. Wilson, "FEC Schemes for ATM Traffic over Wireless Links," *IEEE MILCOM'96*, vol. 3, pp. 948-953, 1996.
- [21] Y. Nakayama and S. Aikawa, " Cell Discard and TDMA Synchronization Using FEC in Wireless ATM Systems," *IEEE Journal On Selected Areas In Comm.*, vol. 15, No. 1, pp 129-34, January 1997.
- [22] M. Al-Khatib and M. Bayoumi, "New Automatic Error Control Systems With ATM Cell Header Reduction for Wireless ATM Networks," *Proc. of IEEE*, pp. 233-238, 1999.
- [23] D. Moore and M. Rice, " Variable Rate Error Control for Wireless ATM Networks," *IEEE ICC'95*, vol. 2, pp. 988-992, 1995.
- [24] D. Raychaudhur, " ATM based Transport Architecture for Multi-service Wireless Communications Networks", *IEEE ICC'94*, vol. 1, pp. 559-565, 1994
- [25] S. Aikawa, Y. Motoyama, and M. Umehira, " Forward Error Correction Schemes for Wireless ATM Systems," *IEEE ICC'96*, vol. 1, pp. 454-458, 1996
- [26] M. Chiani and A. Volta, "Hybrid ARQ/FEC Techniques for Wireless ATM Local Area Networks," *IEEE PIMRC'96*, vol. 3, pp. 898-902, 1996
- [27] A. Andereadis, G Benelli, and D. Sennati, " New System for Interconnection of ATM Networks via Wireless Links," *IEE Proc. Communication*, vol. 145, No. 4, pp. 259-264 August 1998.
- [28] H. C. LEE, Y. W. jung, and B. S. Lee, "Performance improvement of ATM Data Transmission over Wireless Links," *IEEE ICCT'98*, vol. 2, pp. 1-8, October 1998
- [29] F. Borgonovo, and A. Capone, " Comparison of Different Error Control Schemes for Wireless ATM," *IEEE WCNC'99*, vol.1, pp. 466-470, 1999



- [30] H. Ohta and T. Kitami, "A Cell Loss Recovery Method Using FEC in ATM Networks," *IEEE Journal on Selected Areas in Communication*, vol.9, pp. 1471-1483, December 1991.
- [31] A.K. Elhakeem, M. A. Kousa and H. Yang, "A hybrid Multilayer Error Control Technique for multihop ATM Network," 1998, *IEEE International Conference on Communication (ICC 98)* vol. 2, pp. 1168-1173.
- [32] M. A. Kousa, A. K. Elhakeem, and H. Yang, "Performance of ATM Networks Under Hybrid ARQ/FEC Error Control Scheme," *IEEE/ACM Transactions on Networking*, vol.7, December 1999, pp. 917-925.
- [33] M. A. Kousa and A. H. Mugaibel, "Cell Loss Recovery Using Two-Dimensional Erasure Correction for ATM Networks," *7<sup>th</sup> International Conference on Telecommunication Systems*, March 1999, pp. 85-89.
- [34] S. Hazem, *A Hybrid Error Control Scheme for ATM Networks*, MS. Thesis, King Fahd University, Saudi Arabia, May 2000.
- [35] E. Ayanoglu and M. Oguz, "Performance Analysis of Two-level Forward Error Correction for Lost Cell Recovery in ATM Networks," in *Proc. IEEE INFOCOM'95*, vol. 23, pp 723-737, April 1995.
- [36] D. X. Chen, and J. W. Mark, "Performance Analysis of Output Buffered Fast Packet switches with Bursty traffic Loading," *IEEE GLOBECOM'91*, pp. 455-459, 1991.
- [37] M. G. Hluchyj, and M. J. Karol, "Queueing in High-Performance Packet Switching," *IEEE Journal on Selected Areas in Commun.*, vol. 6 , No. 9, pp. 1587-1597, Dec. 1988.

## **Vita**

Mahfooz Saleh Bin Mahfooz was born in Saudi Arabia. He obtained a B. Sc. Degree in Electrical Engineering from Al-ISRA University, Jordan, in 1998. He joined King Fahd University of Petroleum and Minerals (KFUPM) as graduate student in 1998. His successful defence of this thesis at KFUPM in May, 2001 marks His acquisition of a Masters of Science degree in Electrical engineering (Communication Option).